# CH579 Datasheet

V2.1

http://wch.cn

## Overview

CH579 is an ARM 32-bit microcontroller that integrates BLE wireless communication. The chip has integrated rich peripheral resources such as Bluetooth Low Energy (BLE) communication module, Ethernet controller and transceiver, full-speed USB host/device controller and transceiver, segment LCD driver module, ADC, TouchKey detection module and RTC, etc.

## Features

- **Core:**

- ARM 32-bit Cortex-M0 core
- 40MHz maximum frequency, 32KHz minimum frequency

- **257K bytes non-volatile memory FlashROM:**

- 250KB user application memory area, CodeFlash
- 2KB user non-volatile data memory area, DataFlash
- 4KB system bootloader program memory area, BootLoader
- 1KB system non-volatile configuration information memory area, InfoFlash
- Support ICP, ISP and IAP, support OTA wireless upgrade

- **32K bytes volatile data storage SRAM:**

- 16KB memory area only supplied by the main power source, RAM16K
- 14KB sleep retention memory area supplied by two power sources, RAM14K
- 2KB sleep retention memory area supplied by two power sources, RAM2K

- **Power supply management and low-power consumption:**

- Support 3.3V and 2.5V power supply; a range between 2.2V and 3.6V
- Built-in DC-DC conversion, 6mA current when transmitting power at 0dBm
- Idle mode: 1.2mA
- Halt mode: 420uA
- Sleep mode: 0.6uA~2.6uA, multi-gear
- Shutdown mode: 0.2uA~1.5uA, multi-gear
- Low-voltage monitoring of battery voltage

- **Security features:** AES-128 encryption and decryption, unique chip ID
- **Bluetooth Low Energy (BLE):**

- Integrated 2.4GHz RF transceiver, baseband and link control
- Single-ended RF interface, no external inductor required, simplified board design
- Receive sensitivity of -93dBm, programmable+3dBm transmit power
- BLE is compliant with the specification of Bluetooth Low Energy 4.2
- Wireless communication distance is about 100m at 0dBm transmit power
- Optional PCB on board antenna in various shapes and sizes
- Provide optimized protocol stack and application layer API, support networking

- **Ethernet:**

- Built-in MAC controller with DMA support
- Integrated 10Mbps transceiver PHY, built-in 50Ω matching resistor
- Communication distance 200 m, support Auto-Negotiation, support energy saving

- **Universal Serial Bus (USB):**

- Built-in USB controller and DMA, support 64-byte data packet
- Integrated USB 2.0 full-speed transceiver PHY, no peripherals required
- Support full/low-speed Host and Device modes
- Support USB type-C master/slave current detection

- **Real-time clock (RTC):** Support two modes: timing and trigger
- **Segment LCD:** Support 96-point (24×4) LCD panel
- **Analog to digital conversion (ADC):**

- 12-bit analog-to-digital converter, support differential and single-ended inputs
- 14 external analog signal channels and 2-channel internal signals

- **TouchKey detection module:** 14 channels
- **Timer and pulse width modulation (PWM):**

- 4 groups of 26-bit timers, 16MHz main frequency, timing up to 4.2S
- 4 channels of capture / sampling, support for rising / falling / both sides
- 4 channels of 26-bit PWM output, 8 channels of 8-bit PWM output

- **Universal Asynchronous Receiver/Transmitter (UART):**

- 4 groups of independent UARTs, compatible with 16C550, built-in 8-level FIFO
- 23-bit counter, up to maximum baud rate of 5Mbps
- UART0 supports Modem and hardware automatic flow control
- UART0 supports slave address matching automatically while communicating with multi-machine.

- **Serial Peripheral Interface (SPI):**

- 2 groups of independent SPIs, built-in FIFO
- The frequency of SCK serial clock is up to half of the system's main frequency

- SPI0 supports the modes of Master and Slave, supports DMA

- **LED dot matrix interface:** Support 1/2/4-channel data line
- **Clock:** Built-in 32MHz and 32KHz clocks, built-in PLL
- **8-bit passive parallel port**
- **Temperature sensor (TS)**
- **General Purpose Input/Output (GPIO):**

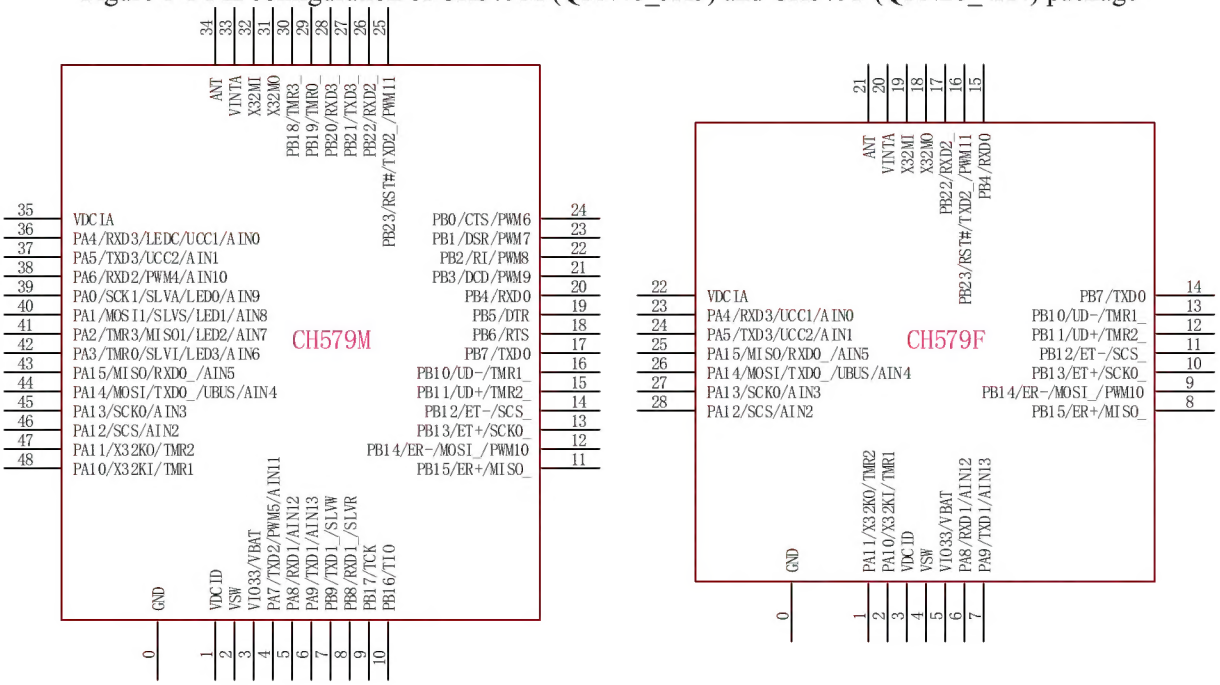
- 40 GPIOs, 4 of which support 5V signal input
- 32 interrupt inputs, 32 wake-up inputs

- **Package form:** QFN48_5×5, QFN28_4×4

# Chapter 1 Pin Information

## 1.1 Pin arrangement

Figure 1-1 Pin configuration of CH579M (QFN48_5X5) and CH579F (QFN28_4X4) package




## 1.2 Pin definitions

| QFN48 Pin No. | QFN28 Pin No. | Pin name | Pin type | Multiplexing feature prioritized | Function description |
|---|---|---|---|---|---|
| 0 | 0 | GND | Power | - | Common ground, the reference point of voltage is 0V. |
| 1 | 3 | VDCID | Power | - | Power input of the internal digital circuit LDO regulator, requires an external decoupling capacitor.<br>2.2uF is recommended when DC-DC is enabled, and not less than 0.1uF is recommended when not enabled. |
| 2 | 4 | VSW | Power | - | Internal DC-DC switch output, a 22uH or 33uH inductor connected to the VDCID must be close to the pin in<br>serious when DC-DC is enabled. It can be directly connected to the VDCID when DC-DC is not enabled. |
| 3 | 5 | VIO33 | Power | VBAT | I/O and DC-DC or battery power input, an external decoupling capacitor shall be close to the pin. 2.2uF is recommended when DC-DC is enabled, and not less<br>than 1uF is recommended when DC-DC is not enabled. |
| 4 | None | PA7 | I/O/A | TXD2<br>/PWM5<br>/AIN11 | PA7: General-purpose bidirectional digital I/O pin.<br>TXD2: UART2 serial data output.<br>PWM5: Pulse width modulation output channel 5.<br>AIN11: ADC analog signal input channel 11. |
| 5 | 6 | PA8 | I/O/A | RXD1<br>/AINT12 | PA8: General-purpose bidirectional digital I/O pin.<br>RXD1: UART1 serial data input.<br>AIN12: ADC analog signal input channel 12. |
| 6 | 7 | PA9 | I/O/A | TXD1<br>/AINT13 | PA9: General-purpose bidirectional digital I/O pin.<br>TXD1: UART1 serial data output.<br>AIN13: ADC analog signal input channel 13. |
| 7 | None | PB9 | I/O | TXD1_ | PB9: General-purpose bidirectional digital I/O pin. |

| | | | | | |
|---|---|---|---|---|---|
| | | | | /SLVW | TXD1_: TXD1 pin mapping of UART1.<br>SLVW: Write control signal input of passive parallel port, active at low level. |
| 8 | None | PB8 | I/O | RXD1_<br>/SLVR | PB8: General-purpose bidirectional digital I/O pin.<br>RXD1_: RXD1 pin mapping of UART1.<br>SLVR: Read control signal input of passive parallel port, active at low level. |
| 9 | None | PB17 | I/O | TCK | PB17: General-purpose bidirectional digital I/O pin.<br>TCK: Serial clock input of the two-wire emulation debug interface. |
| 10 | None | PB16 | I/O | TIO | PB16: General-purpose bidirectional digital I/O pin.<br>TIO: Serial data input and output of the two-wire emulation debug interface, with built-in pull-up. |
| 11 | 8 | PB15 | I/O/A<br>/5VT | ER+<br>/MISO_ | PB15: General-purpose bidirectional digital I/O pin. ER+: Ethernet receives RX+ signals.<br>MISO_: MISO pin mapping of SPI0. |
| 12 | 9 | PB14 | I/O/A<br>/5VT | ER-<br>/MOSI_<br>/PWM10 | PB14: General-purpose bidirectional digital I/O pin.<br>ER-: Ethernet receives RX- signals.<br>MOSI_: MOSI pin mapping of SPI0.<br>PWM10: Pulse width modulation output channel 10. |
| 13 | 10 | PB13 | I/O/A<br>/5VT | ET+<br>/SCK0_ | PB13: General-purpose bidirectional digital I/O pin.<br>ET+: Ethernet sends TX+ signals.<br>SCK0_: SCK pin mapping of SPI0. |
| 14 | 11 | PB12 | I/O/A<br>/5VT | ET-<br>/SCS_ | PB12: General-purpose bidirectional digital I/O pin.<br>ET-: Ethernet sends TX- signals.<br>SCS_: SCS pin mapping of SPI0. |
| 15 | 12 | PB11 | I/O/A | UD+<br>/TMR2_ | PB11: General-purpose bidirectional digital I/O pin.<br>UD+: D+ data line of USB bus.<br>TMR2_: TMR2 pin mapping of Timer 2. |
| 16 | 13 | PB10 | I/O/A | UD-<br>/TMR1_ | PB10: General-purpose bidirectional digital I/O pin.<br>UD-: D- data line of USB bus.<br>TMR1_: TMR1 pin mapping of Timer 1. |
| 17 | 14 | PB7 | I/O | TXD0 | PB7: General-purpose bidirectional digital I/O pin.<br>TXD0: UART0 serial data output. |
| 18 | None | PB6 | I/O | RTS | PB6: General-purpose bidirectional digital I/O pin.<br>RTS: MODEM output signal of UART0, request to send. |
| 19 | None | PB5 | I/O | DTR | PB5: General-purpose bidirectional digital I/O pin.<br>DTR: MODEM output signal of UART0, data terminal ready. |
| 20 | 15 | PB4 | I/O | RXD0 | PB4: General-purpose bidirectional digital I/O pin.<br>RXD0: UART0 serial data input. |
| 21 | None | PB3 | I/O | DCD<br>/PWM9 | PB3: General-purpose bidirectional digital I/O pin.<br>DCD: MODEM input signal of UART0, data carrier detection.<br>PWM9: Pulse width modulation output channel 9. |
| 22 | None | PB2 | I/O | RI<br>/PWM8 | PB2: General-purpose bidirectional digital I/O pin.<br>RI: MODEM input signal of UART0, ringing indication.<br>PWM8: Pulse width modulation output channel 8. |
| 23 | None | PB1 | I/O | DSR<br>/PWM7 | PB1: General-purpose bidirectional digital I/O pin.<br>DSR: MODEM input signal of UART0, data device ready.<br>PWM7: Pulse width modulation output channel 7. |
| 24 | None | PB0 | I/O | CTS<br>/PWM6 | PB0: General-purpose bidirectional digital I/O pin.<br>CTS: MODEM input signal of UART0, clear to send.<br>PWM6: Pulse width modulation output channel 6. |
| 25 | 16 | PB23 | I/O | RST#<br>/TXD2_<br>/PWM11 | PB23: General-purpose bidirectional digital I/O pin.<br>RST#: External reset input, active at low level, built-in pull-up resistor.<br>TXD2_: TXD2 pin mapping of UART2.<br>PWM11: Pulse width modulation output channel 11. |

| | | | | | |
|---|---|---|---|---|---|
| 26 | 17 | PB22 | I/O | RXD2_ | PB22: General purpose bidirectional digital I/O pin.<br>RXD2_: RXD2 pin mapping of UART2. |
| 27 | None | PB21 | I/O | TXD3_ | PB21: General purpose bidirectional digital I/O pin.<br>TXD3_: TXD3 pin mapping of UART3. |
| 28 | None | PB20 | I/O | RXD3_ | PB20: General purpose bidirectional digital I/O pin.<br>RXD3_: RXD3 pin mapping of UART3. |
| 29 | None | PB19 | I/O | TMR0_ | PB19: General purpose bidirectional digital I/O pin.<br>TMR0_: TMR0 pin mapping of Timer 0. |
| 30 | None | PB18 | I/O | TMR3_ | PB18: General purpose bidirectional digital I/O pin.<br>TMR3_: TMR3 pin mapping of Timer 3. |
| 31 | 18 | X32MO | I/A | - | Inverted output of the high frequency oscillator HSE, it is externally connected to one end of 32 MHz crystal. |
| 32 | 19 | X32MI | A | - | Input of the high frequency oscillator HSE, it is externally connected to the other end of 32MHz crystal. |
| 33 | 20 | VINTA | Power | - | Power supply node of the internal analog circuit, an external decoupling capacitor shall be close to the pin, 2.2uF is recommended (It can be 1uF when DC-DC is not enabled, slightly saving more power but reducing BLE sensitivity). |
| 34 | 21 | ANT | A | - | RF signal input and output, it is recommended to be connected with the antenna directly. |
| 35 | 22 | VDCIA | Power | - | Power input of the internal analog circuit LDO regulator, an external decoupling capacitor is required.<br>It is recommended not to be less than 0.1uF. It is recommended to directly connect to the VDCID. |
| 36 | 23 | PA4 | I/O/A | RXD3<br>/LEDC<br>/UCC1<br>/AIN0 | PA4: General-purpose bidirectional digital I/O pin.<br>RXD3: UART3 serial data input.<br>LEDC: LED screen interface serial clock output.<br>UCC1: USB type-C bidirectional configuration channel 1.<br>AIN0: ADC analog signal input channel 0. |
| 37 | 24 | PA5 | I/O/A | TXD3<br>/UCC2<br>/AIN1 | PA5: General-purpose bidirectional digital I/O pin.<br>TXD3: UART3 serial data output.<br>UCC2: USB type-C bidirectional configuration channel 2.<br>AIN1: ADC analog signal input channel 1. |
| 38 | None | PA6 | I/O/A | RXD2<br>/PWM4<br>/AIN10 | PA6: General-purpose bidirectional digital I/O pin.<br>RXD2: UART2 serial data input.<br>PWM4: Pulse width modulation output channel 4.<br>AIN10: ADC analog signal input channel 10. |
| 39 | None | PA0 | I/O/A | SCK1<br>/SLVA<br>/LED0<br>/AIN9 | PA0: General-purpose bidirectional digital I/O pin.<br>SCK1: SPI1 serial clock output.<br>SLVA: Address input of passive parallel ports, select command and status port when at high level, and select data port when at low level.<br>LED0: LED screen interface serial data output 0.<br>AIN9: ADC analog signal input channel 9. |
| 40 | None | PA1 | I/O/A | MOSI1<br>/SLVS<br>/LED1<br>/AIN8 | PA1: General-purpose bidirectional digital I/O pin.<br>MOSI1: SPI1 serial data output.<br>SLVS: Chip select control input for passive parallel port, active at low level.<br>LED1: LED screen interface serial data output 1.<br>AIN8: ADC analog signal input channel 8. |
| 41 | None | PA2 | I/O/A | TMR3<br>/MISO1<br>/LED2<br>/AIN7 | PA2: General-purpose bidirectional digital I/O pin.<br>TMR3: Capture input 3 and PWM output channel 3 of Timer 3.<br>MISO1: SPI1 serial data input, reused for output in 2- wire mode.<br>LED2: LED screen interface serial data output 2.<br>AIN7: ADC analog signal input channel 7. |
| 42 | None | PA3 | I/O/A | TMR0 | PA3: General-purpose bidirectional digital I/O pin. |

| | | | | | |
|---|---|---|---|---|---|
| | | | | /SLVI<br>/LED3<br>/AIN6 | TMR0: Capture input 0 and PWM output channel 0 of Timer 0.<br>SLVI: Interrupt request output of passive parallel port, active at low level.<br>LED3: LED screen interface serial data output 3.<br>AIN6: ADC analog signal input channel 6. |
| 43 | 25 | PA15 | I/O/A | MISO<br>/RXD0_<br>/AIN5 | PA15: General-purpose bidirectional digital I/O pin.<br>MISO: SPI0 serial data pin, master input/slave output.<br>RXD0_: RXD0 pin mapping of UART0.<br>AIN5: ADC analog signal input channel 5. |
| 44 | 26 | PA14 | I/O/A | MOSI<br>/TXD0_<br>/UBUS<br>/AIN4 | PA14: General-purpose bidirectional digital I/O pin.<br>MOSI: SPI0 serial data pin, master output / slave input.<br>TXD0_: TXD0 pin mapping of UART0.<br>UBUS: USB type-C bus voltage detection input.<br>AIN4: ADC analog signal input channel 4. |
| 45 | 27 | PA13 | I/O/A | SCK0<br>/AIN3 | PA13: General-purpose bidirectional digital I/O pin.<br>SCK0: SPI0 serial clock pin, master output / slave input.<br>AIN3: ADC analog signal input channel 3. |
| 46 | 28 | PA12 | I/O/A | SCS<br>/AIN2 | PA12: General-purpose bidirectional digital I/O pin.<br>SCS: Chip select input in the SPI0 slave mode, active at low level.<br>AIN2: ADC analog signal input channel 2. |
| 47 | 1 | PA11 | I/O/A | X32KO<br>/TMR2 | PA11: General-purpose bidirectional digital I/O pin.<br>X32KO: The inverted output end of the low frequency oscillator is externally connected to one end of the 32KHz crystal.<br>TMR2: Capture input 2 and PWM output channel 2 of Timer 2. |
| 48 | 2 | PA10 | I/O/A | X32KI<br>/TMR1 | PA10: General-purpose bidirectional digital I/O pin.<br>X32KI: The input end of the low frequency oscillator is externally connected to the other end of the 32KHz crystal.<br>TMR1: Capture input 1 and PWM output channel 1 of Timer 1. |

Pin descriptions for reuse of segmented LCD drivers

| Pin name | Type | Alternative function | Function description |
|---|---|---|---|
| PA0～PA3 | A | COM0～COM3 | Drive each common end of the segment LCD, and select some or all of them as needed. |
| PB0～PB23 | A | SEG0～SEG23 | Drive each segment of the segment LCD, and select some or all of them as needed.<br>There is 6-bit RB_PIN_SEG*_IE in the R16_PIN_ ANALOG_IE register. Each bit controls 4 segment pins at the same time. When the bit is 0, 4 pins are used for digital input or other non-LCD features. When the bit is 1, 4 pins are used for LCD segment drive. |

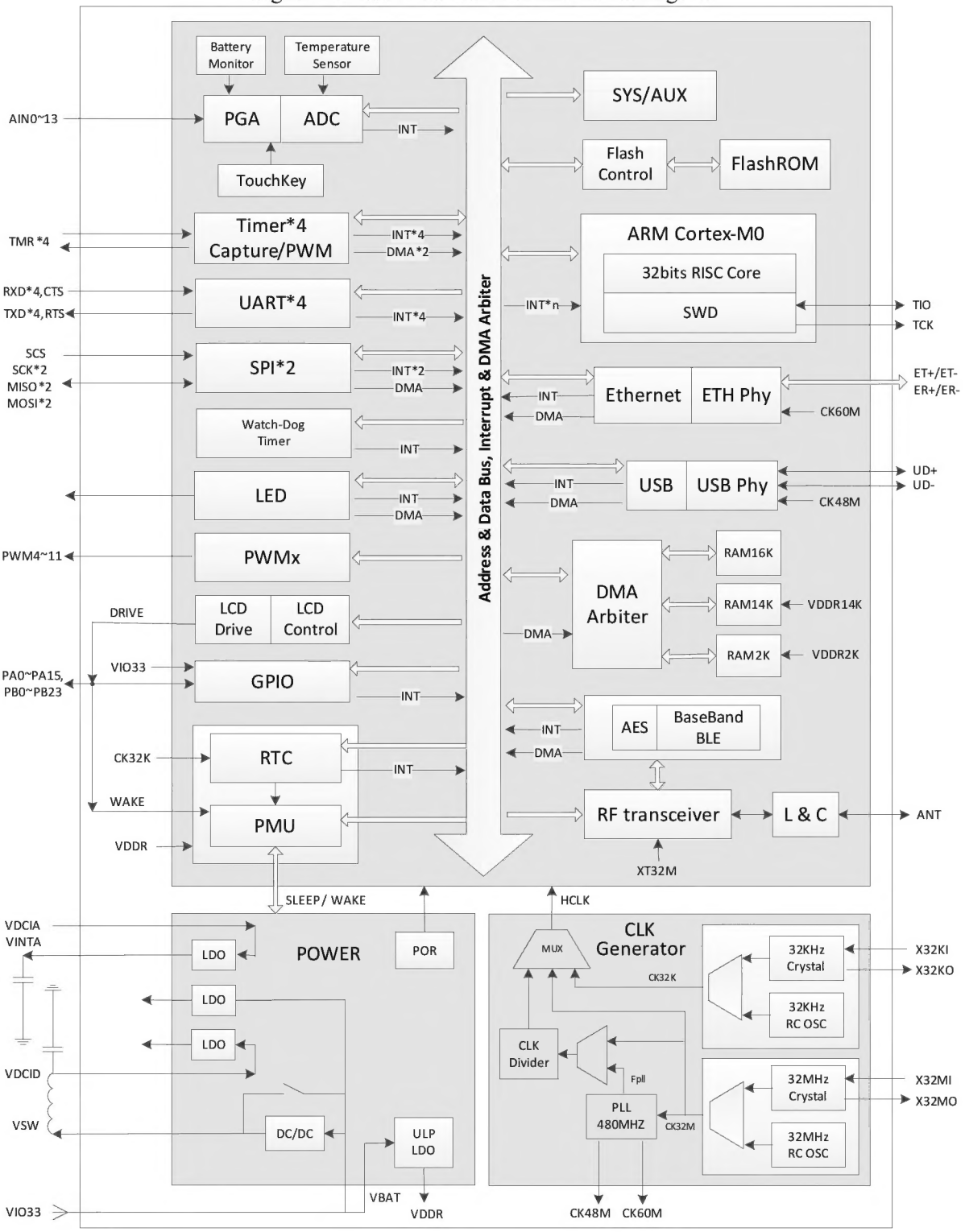
Note:
(1) Pin type:
I = TTL / CMOS level Schmidt input;
O = CMOS level three-state output;
A = Analog signal input or output;
5VT=Support 5V signal voltage input.
(2) The multiplexing feature and the mapping of pins are arranged in the table according to their priorities from the highest to the lowest. Among them, the GPIO feature is the lowest priority.

# Chapter 2 System Architecture and Memory

## 2.1 System architecture

The figure below illustrates the block diagram of the structure of the CH579 chip system.

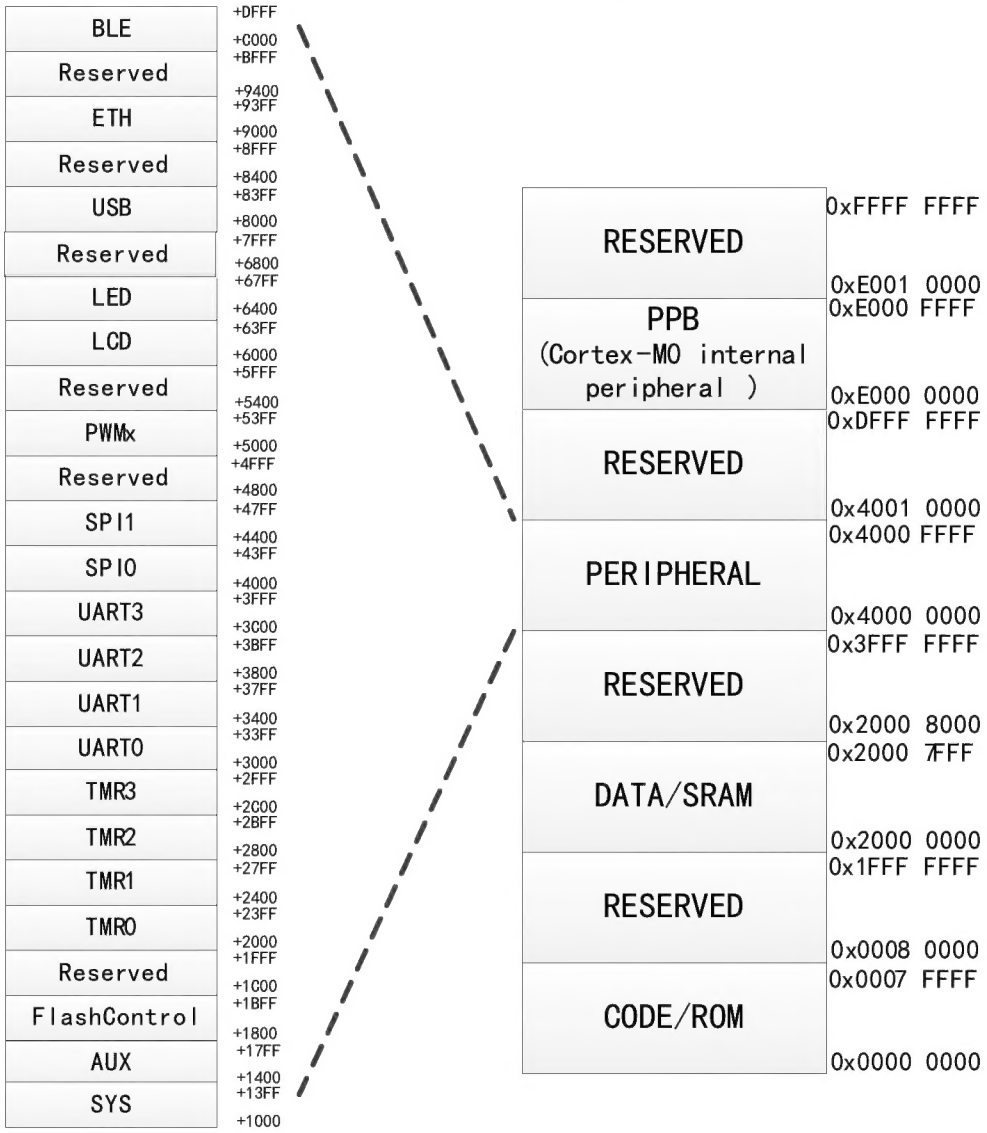
Figure 2-1 CH579 internal structure block diagram



## 2.2 Memory mapping

The addressing space of CH579 mainly includes several different areas such as CODE area/FlashROM, DATA area/SRAM and peripherals, as shown in the figure below.

Figure 2-2 Memory mapping diagram




## 2.3 Memory mapping table

The address range of each memory mapping area is indicated in the following table:

Table 2-1 Addresses in the memory mapping area

| Address range | Application | Description |
|---|---|---|
| 0x00000000-0x0007FFFF | On-chip CODE area, non-volatile memory | 257KB，FlashROM |
| 0x00080000-0x1FFFFFFF | Reserved | - |
| 0x20000000-0x20007FFF | On-chip DATA area, volatile memory | 32KB，SRAM |
| 0x20008000-0x3FFFFFFF | Reserved | - |
| 0x40000000-0x4000FFFF | Various peripherals | Multiple peripheral modules |
| 0x40010000-0xDFFFFFFF | Reserved | - |
| 0xE0000000-0xE000FFFF | PPB | Internal peripheral bus area |
| 0xE0010000-0xFFFFFFFF | Reserved | - |

### 2.3.1 On-Chip CODE area mapping table

Table 2-2 Addresses in the CODE Area

| Address range | Application | Description |
|---|---|---|
| 0x00000000-0x0003E7FF | User application program memory area, CodeFlash | 250KB |
| 0x0003E800-0x0003EFFF | User non-volatile data memory area, DataFlash | 2KB |
| 0x0003F000-0x0003FFFF | System bootloader memory area, BootLoader | 4KB |
| 0x00040000-0x000403FF | System non-volatile configuration information memory area, InfoFlash | 1KB |
| 0x00040400-0x0007FFFF | Reserved | - |

The configuration information on address 0x00040010 can be set by users through tools.

Table 2-3 User-level non-volatile configuration information

| Bit address | Name | Application | Default value |
|---|---|---|---|
| Bit 2～Bit 0 | RESERVED | Reserved | 000b |
| Bit 3 | CFG_RESET_EN | RST# External manual reset input pin enable | 0 |
| Bit 4 | CFG_DEBUG_EN | 2-wire emulation debug interface enable | 1 |
| Bit 5 | RESERVED | Reserved | 0 |
| Bit 6 | CFG_BOOT_EN | System BootLoader enable | 1 |
| Bit 7 | CFG_ROM_READ | Code and data protection mode in FlashROM:<br>0—Disable the programmer to readout and program is confidential;<br>1—Allow readout | 1 |
| Bit 27~Bit 8 | RESERVED | Reserved | 0FFFFh |
| Bit 31~Bit 28 | VALID_SIG | Valid indication of configuration information, a fixed value | 0101b |

### 2.3.2 On-Chip DATA area mapping table

Table 2-3 DATA area address

| Address range | Application | Description |
|---|---|---|
| 0x20000000-0x20003FFF | Memory area only supplied by the main power source, RAM16K | 16KB |
| 0x20004000-0x200077FF | Independently maintained memory area supplied by the main + auxiliary power source, RAM14K | 14KB |
| 0x20007800-0x20007FFF | Independently maintained memory area supplied by the main + auxiliary power source, RAM2K | 2KB |

### 2.3.3 Peripheral address assignment

CH579 mainly includes the following peripherals. Each peripheral occupies a certain address space. The actual access address of the peripheral registers is: base address + offset address. In the following sections, the address of the register is described in detail. The following table indicates the assignment table for each peripheral base address.

Table 2-5 Assignment table for peripheral base addresses

| Peripheral No. | Peripheral name | Peripheral base address |
|---|---|---|
| 1 | SYS (PMU/RTC/GPIO, etc.) | 0x4000 1000 |
| 2 | AUX (ADC/TKEY/PLL, etc.) | 0x4000 1400 |
| 3 | FlashROM-Control | 0x4000 1800 |
| 4 | TMR0 | 0x4000 2000 |
| 5 | TMR1 | 0x4000 2400 |
| 6 | TMR2 | 0x4000 2800 |
| 7 | TMR3 | 0x4000 2C00 |
| 8 | UART0 | 0x4000 3000 |

| | | |
|---|---|---|
| 9 | UART1 | 0x4000 3400 |
| 10 | UART2 | 0x4000 3800 |
| 11 | UART3 | 0x4000 3C00 |
| 12 | SPI0 | 0x4000 4000 |
| 13 | SPI1 | 0x4000 4400 |
| 14 | PWMx(PWM4～PWM11) | 0x4000 5000 |
| 15 | LCD | 0x4000 6000 |
| 16 | LED | 0x4000 6400 |
| 17 | USB | 0x4000 8000 |
| 18 | ETH | 0x4000 9000 |
| 19 | Radio:BLE | 0x4000 C000<br>0x4000 D000 |

The following table provides an explanation of "access" of the register described in the following sections:

Table 2-6 Description of access attributes

| Abbreviation | Description |
|---|---|
| RF | Read-only. The read-out value is fixed and is not affected by reset. |
| RO | Read-only. |
| WO | Write-only. The read-out value is 0 or invalid. |
| RZ | Read-only. It is cleared to 0 automatically after reading. |
| WZ | It is cleared to 0 after writing. |
| RW | Readable and writable. |
| RW1 | Readable. It is cleared to 0 after writing 1. |
| WA | Write-only. The read-out value is 0 or invalid only in the safe mode. |
| RWA | Readable. Writable only in the safe mode. |

The following table explains the abbreviations used in the subsequent sections:

Table 2-7 Description of noun abbreviations

| Abbreviation | Description |
|---|---|
| HSE | External high frequency crystal oscillator clock source (32MHz recommended) |
| HSI | Internal high frequency RC clock oscillation source (32MHz after factory calibration) |
| LSE | External low frequency crystal oscillator clock source (32KHz recommended) |
| LSI | Internal low frequency RC clock oscillation source (32KHz after calibration when the application software is running) |
| CK32M | High frequency clock source (32MHz at default) |
| CK32K | Low frequency clock source (32KHz at default) |
| Fpll | PLL output clock (the default frequency is 480MHz) |
| HCLK | System main frequency clock |
| Fsys | Frequency of system main-frequency clock |
| Tsys | Cycle of system main-frequency clock (1/Fsys) |
| RAM2K | 2KB SRAM of the highest address |
| RAM14K | 14KB SRAM of the next highest address |
| 0x | The data starting with it represents a hexadecimal number |
| H | The data ending with it represents a hexadecimal number |
| B | The data ending with it represents a binary number |

# Chapter 3 Interrupt

## 3.1 Interrupt controller

The system supports up to 20 sets of interrupt signal sources. Each interrupt request has independent trigger and disabling control bits as well as dedicated status bits.

## 3.2 SysTick calibration value

When the SysTick clock is set to 32MHz, the SysTick calibration value is fixed at 32,000, which will generate a time reference of 1ms.

## 3.3 Interrupt and exception vectors

The following table lists the vector table of the chip system.

Table 3-1 Interrupt vectors

| Position | Priority | Priority type | Name | Description | Address |
|---|---|---|---|---|---|
| | - | - | - | Reserved | 0x0000_0000 |
| -15 | -3 | Fixed | Reset | Reset | 0x0000_0004 |
| -14 | -2 | Fixed | NMI | Non-maskable interrupts | 0x0000_0008 |
| -13 | -1 | Fixed | Hard fault | Failures of all types | 0x0000_000C |
| | - | - | - | Reserved | - |
| -5 | 0 | Settable | SVCall | System service call via SWI instruction | 0x0000_002C |
| | - | - | - | Reserved | - |
| -2 | 1 | Settable | PendSV | Suspend system service | 0x0000_0038 |
| -1 | 2 | Settable | SysTick | SysTick timer | 0x0000_003C |
| 0 | 3 | Settable | TMR0 | TMR0 Timer 0 interrupt | 0x0000_0040 |
| 1 | 4 | Settable | GPIO | GPIO general-purpose I/O interrrupt | 0x0000_0044 |
| 2 | 5 | Settable | SLAVE | Passive parallel port interrupt | 0x0000_0048 |
| 3 | 6 | Settable | SPI0 | SPI0 interrupt | 0x0000_004C |
| 4 | 7 | Settable | BLEL | LLE interrupt of wireless module | 0x0000_0050 |
| 5 | 8 | Settable | BLEB | BB interrupt of wireless module | 0x0000_0054 |
| 6 | 9 | Settable | USB | USB interrupt | 0x0000_0058 |
| 7 | 10 | Settable | ETH | Ethernet interrupt | 0x0000_005C |
| 8 | 11 | Settable | TMR1 | TMR1 Timer 1 interrupt | 0x0000_0060 |
| 9 | 12 | Settable | TMR2 | TMR2 Timer 2 interrupt | 0x0000_0064 |
| 10 | 13 | Settable | UART0 | UART0 Asynchronous serial port 0 interrupt | 0x0000_0068 |
| 11 | 14 | Settable | UART1 | UART1 Asynchronous serial port 1 interrupt | 0x0000_006C |
| 12 | 15 | Settable | RTC | RTC real-time clock interrupt | 0x0000_0070 |
| 13 | 16 | Settable | ADC | ADC interrupt | 0x0000_0074 |
| 14 | 17 | Settable | SPI1 | SPI1 interrupt | 0x0000_0078 |
| 15 | 18 | Settable | LED | LED screen control interrupt | 0x0000_007C |
| 16 | 19 | Settable | TMR3 | TMR3 Timer 3 interrupt | 0x0000_0080 |
| 17 | 20 | Settable | UART2 | UART2 Asynchronous serial port 2 interrupt | 0x0000_0084 |
| 18 | 21 | Settable | UART3 | UART3 Asynchronous serial port 3 interrupt | 0x0000_0088 |
| 19 | 22 | Settable | WDOG_BAT | Watchdog timer interrupt/battery low voltage interrupt | 0x0000_008C |

# Chapter 4 System control

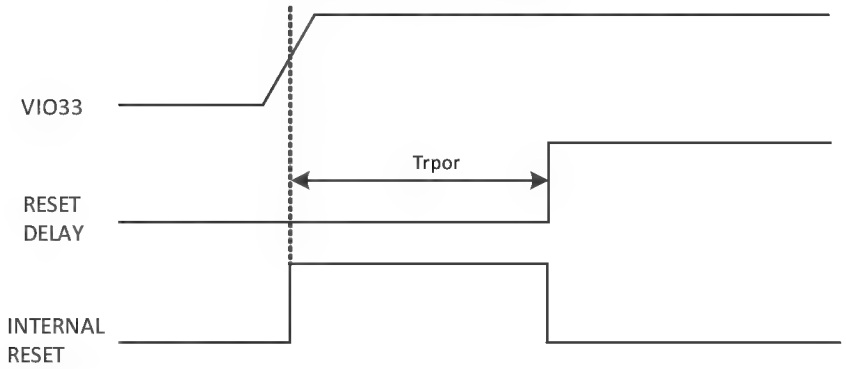
## 4.1 Reset control

The system supports 6 forms of reset: RPOR (real power on reset), external MR (manual reset), internal SR (software reset), WTR (watch-dog time-out reset), GRWSM (global reset by waking under shutdown mode) and LRW (local reset by waking).
Register R8_GLOB_RESET_KEEP and register RB_ROM_CODE_OFS are only reset when RPOR and GRWSM occur, and they are not affected by other forms of reset.
For the timing parameters and reset characteristic parameters in the figure below, please refer to the timing parameter table in Section 20.5.

### 4.1.1 Real Power on Reset (RPOR)

When power is turned on, power-on reset is generated inside the chip and delayed to wait for the power supply to stabilize. The figure below illustrates the process of power-on reset.

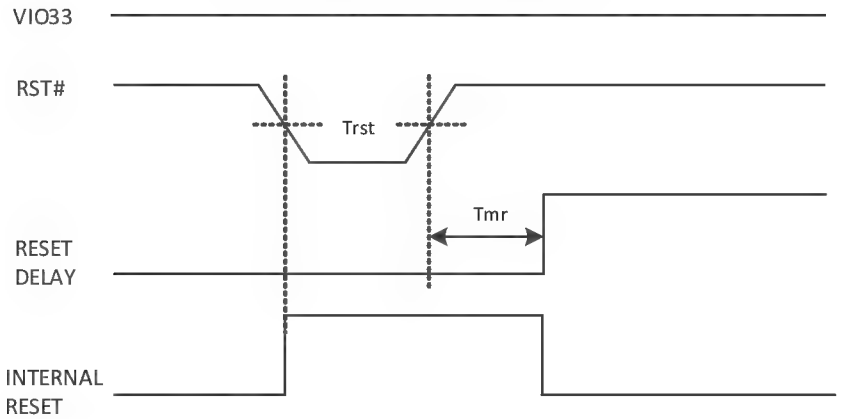
Figure 4-1 Real power on reset




### 4.1.2 External Manual Reset (MR)

External manual reset is triggered by a low level externally applied to the RST# pin. When the reset low level duration is greater than the minimum reset pulse width (Trst), the system is triggered to reset.

Figure 4-2 External manual reset




### 4.1.3 Internal Software Reset (SR)

Internal software reset is used to automatically reset without external intervention. Software reset can be implemented by setting the bit RB_SOFTWARE_RESET of the global reset configuration register (R8_RST_WDOG_CTRL) to '1'. This bit is automatically cleared to 0.

### 4.1.4 Watchdog Timeout Reset (WTR)

The watchdog feature is based on an 8-bit count-up counter with a count clock cycle of 131072/Fsys. When the feature of watchdog timeout reset is turned on, the entire system will be reset once the counter overflows.

### 4.1.5 Watchdog Timeout Reset (WTR)

Once the system enters the shutdown mode (see the chapter of Power Management for details), the system will perform the wake-up operation in an orderly manner under the action of wake-up signals. After wakeup, the system will perform a global reset. The effect of this reset is similar to that of power-on reset.

### 4.1.6 Local Reset by Waking (LRW)

If the system is woken up from sleep mode, a reset will be generated after the associated power supply is ready. This reset is a partial reset, with a selective reset of the registers that are powered down in the sleep mode as needed. In the sleep mode, the registers of each feature module are divided into three categories:
The first type is key registers (e.g., configuration/mode, etc.) belonging to a feature module that requires data retention. At the time of sleep, the auxiliary power supply continues to supply power, and the data is not lost. Both sleep and wakeup have no effect on their data;
The second type is regenerative registers (e.g., counter, FIFO, etc.) belonging to a feature module that requires data retention. Power is turned off during sleep, and the data is a random number (e.g., a FIFO memory cell) or is reset (e.g., a FIFO counter) after wakeup;
The third type is registers belonging to a feature module that does not require data retention. Power is turned off during sleep, and the data is a random number (e.g., a FIFO memory cell) or is reset after wakeup (e.g., a FIFO counter, configuration/mode register).
LRW is used for the second and third type of reset registers.

## 4.2 Safe access

The properties of some registers of the system are "RWA" or "WA", indicating that the current register can be safely accessed and can be read directly, but write-in needs to enter the secure access mode.
First write in R8_SAFE_ACCESS_SIG register 0x57;
Then write in R8_SAFE_ACCESS_SIG register 0xA8;
At this time, you can enter the mode of safe access and operate the register with the “RWA/WA” attribute. Afterwards, about 16 system main frequency cycles (Tsys) are in the safe mode. One or more security-class registers can be overwritten during the validity period. If the above validity period is exceeded, the security mode will be automatically terminated. Alternatively, the security mode can be terminated in advance by writing in 0x00 to the R8_SAFE_ACCESS_SIG register.

## 4.3 Register description

Table 4-1 List of registers related to system control

| Name | Access address | Description | Default value |
|---|---|---|---|
| R8_SAFE_ACCESS_SIG | 0x40001040 | Safe access flag register | 0x00 |
| R8_CHIP_ID | 0x40001041 | Chip ID register | 0x79 |
| R8_SAFE_ACCESS_ID | 0x40001042 | Safe access ID register | 0x04 |
| R8_WDOG_COUNT | 0x40001043 | Watchdog counter register | 0x00 |
| R8_RESET_STATUS | 0x40001044 | Reset status register | 0x01 |
| R8_GLOB_CFG_INFO | 0x40001045 | Global configuration information status register | 0xEX |
| R8_RST_WDOG_CTRL | 0x40001046 | Watchdog and reset configuration register | 0x00 |
| R8_GLOB_RESET_KEEP | 0x40001047 | Reset hold register | 0x00 |
| R8_CFG_FLASH | 0x4000104A | FlashROM configuration register | 0x0X |
| R32_FLASH_DATA | 0x40001800 | FlashROM data register | 0xXXXXXXXX |
| R32_FLASH_ADDR | 0x40001804 | FlashROM address register | 0xXXXXXXXX |
| R8_FLASH_COMMAND | 0x40001808 | FlashROM command register | 0x00 |
| R8_FLASH_PROTECT | 0x40001809 | FlashROM protection control register | 0x00 |
| R16_FLASH_STATUS | 0x4000180A | FlashROM status register | 0x0X00 |

**Safe Access Flag Register (R8_SAFE_ACCESS_SIG)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7:0] | R8_SAFE_ACCESS_SIG | WO | Secure access flag register.<br>Some registers (access attribute is RWA) are protection registers and must enter the mode of safe access for write operations. Write 0x57 to this register and then write 0xA8 to enter the mode of safe access. The time limit is about 16 main clock cycles (Tsys). If it is exceeded, automatic protection is enabled. You can write any other value to force direct exit from the mode of safe access and return to the protection state. | 00h |
| 7 | Reserved | RO | Reserved | 0 |
| [6:4] | RB_SAFE_ACC_TIMER | RO | Current count of the time of safe access. | 000b |
| 3 | RB_SAFE_ACC_ACT | RO | Status of current safe access mode:<br>1: Writeable in the mode of unlocked/safe access;<br>0: Locked, the RWA attribute register cannot be rewritten. | 0 |
| 2 | Reserved | RO | Reserved | 0 |
| [1:0] | RB_SAFE_ACC_MODE | RO | Status of current safe access mode:<br>11: Safe mode; can be written into attribute RWA register;<br>Other: Non-safe mode. | 00b |

**Chip ID Register (R8_CHIP_ID)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7:0] | R8_CHIP_ID | RF | A fixed value of 79h is used to identify the chip. | 79h |

**Safe Access ID Register (R8_SAFE_ACCESS_ID)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7:0] | R8_SAFE_ACCESS_ID | RF | A fixed value of 04h | 04h |

**Watchdog Counter Register (R8_WDOG_COUNT)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7：0] | R8_WDOG_COUNT | RW | The watchdog counter, which can be preset with the initial value, is automatically incremented and can be cycled from 0xFF to 0x00 and then continue.<br>Count cycle = 131072/Fsys. | 00h |

**Reset Status Register (R8_RESET_STATUS)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7:5] | Reserved | RO | Reserved | 000b |
| 4 | RB_ROM_CODE_OFS | RWA | Select the starting offset address of the user program code in FlashROM. This value is not affected by MR, SR or WTR and can only be cleared when RPOR and GRWSM are valid:<br>0: 0x000000;<br>1: 0x008000 (skip the first 32KB in ROM). | 0 |
| 3 | Reserved | RO | Reserved | 0 |
| [2:0] | RB_RESET_FLAG | RO | Last reset status:<br>000: Software reset SR (when RB_WDOG RST_EN=0, software reset can generate this state, otherwise it can be reset but does not produce this state);<br>001: Real Power On Reset (RPOR); | 001b |

| | | | | |
|---|---|---|---|---|
| | | | 010: Watchdog Timeout Reset (WTR);<br>011: External Manual Reset (MR);<br>101: Global Reset by Waking Under Shutdown Mode (GRWSM);<br>100/110/111: Local Reset by Waking (LRW), and the last reset before is SR/WTR/MR. | |

**Global Configuration Information Status Register (R8_GLOB_CFG_INFO)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7：6] | Reserved | RO | Reserved | 11b |
| 5 | RB_BOOT_LOADER | RO | Bootloader status:<br>1: Currently in the bootloader state;<br>0: Currently in the user program state. | 1/0 |
| 4 | RB_CFG_DEBUG_EN | RO | 2-wire emulation debug interface is enabled:<br>1: Can simulate debugging, can read FlashROM;<br>0: Debugging simulation is disabled. | 0 |
| 3 | RB_CFG_BOOT_EN | RO | The system BootLoader enable status:<br>1: Enabled;<br>0: Not enabled. | 1 |
| 2 | RB_CFG_RESET_EN | RO | RST# external manual reset input enable status:<br>1: Enabled;<br>0: Not enabled. | 0 |
| 1 | Reserved | RO | Reserved | 0 |
| 0 | RB_CFG_ROM_READ | RO | FlashROM code and data area protection status:<br>1: The external programmer is readable;<br>0: Protected, externally inaccessible; the program is kept secret. | 0 |

**Watchdog and Reset Configuration Register (R8_RST_WDOG_CTRL)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7:5] | Reserved | RO | Reserved | 000b |
| 4 | RB_WDOG_INT_FLAG | RW1 | Watchdog timer interrupt flag:<br>1: A watchdog count overflow occurs, that is, R8_WDOG_ COUNT is detected to be incremented from 0xFF to 0x00;<br>0: The watchdog count has not overflowed.<br>For the flag: write 1 to clear, or reload the value of the watchdog counter (R8_WDOG_COUNT) to clear, or execute SEV() to clear. | 0 |
| 3 | Reserved | RO | Reserved | 0 |
| 2 | RB_WDOG_INT_EN | RWA | Watchdog Timer Interrupt Enable bit:<br>1: Enabled, an interrupt is generated after the watchdog count overflows;<br>0: Disable the watchdog timer interrupt. | 0 |
| 1 | RB_WDOG_RST_EN | RWA | Watchdog Timeout Reset Enable bit:<br>1: Enable: the system resets after watchdog count overflows:<br>0: Only as a watchdog timer.<br>*Note: The software reset operation will not affect the status of RB_ RESET_ FLAG after this bit is set to 1.* | 0 |
| 0 | RB_SOFTWARE_RESET | WA/ WZ | System software reset control, automatically cleared after reset: | 0 |

| | | | | |
|---|---|---|---|---|
| | | | 1: Perform system software reset;<br>0: Idle, no action. | |

**Reset Hold Register (R8_GLOB_RESET_KEEP)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7:0] | R8_GLOB_RESET_KEEP | RW | The reset hold register is not affected by manual reset, software reset, watchdog reset, or normal wake-up reset. | 00h |

**FlashROM Configuration Register (R8_CFG_FLASH)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | RB_FLASH_BUSY_EN | RWA | Busy enable bit when FlashROM is acceded in bursts:<br>1: Waiting (recommended); 0: No waiting. | 0 |
| [6:4] | Reserved | RO | Reserved | 000b |
| [3:0] | RB_CFG_FLASH_X | RWA | FlashROM configuration data must be kept unchanged when being written in. | XXXXb |

**FlashROM Data Register (R32_FLASH_DATA)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:0] | R32_FLASH_DATA | RW | FlashROM data register | XXXXXXXXh |

**FlashROM Address Register (R32_FLASH_ADDR)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:0] | R32_FLASH_ADDR | RW | FlashROM address register | XXXXXXXXh |

**FlashROM Command Register (R8_FLASH_COMMAND)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7:0] | R8_FLASH_COMMAND | WO | 0x9A: FlashROM programming operation command;<br>0xA6: FlashROM sector erase operation command. | 00h |

**FlashROM Protection Control Register (R8_FLASH_PROTECT)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7:6] | RB_ROM_WE_MUST_10 | WO | Must write 10b, otherwise it can't be written. | 10b |
| [5:4] | Reserved | RO | Reserved | 00b |
| 3 | RB_ROM_CODE_WE | RW | Erase/Programming Enable bit of FlashROM program memory area CodeFlash:<br>1: Allow erase/programming;<br>0: Erase/programming protection. | 0 |
| 2 | RB_ROM_DATA_WE | RW | Erase/Programming Enable bit of FlashROM program memory area DataFlash:<br>1: Allow erase/programming;<br>0: Erase/programming protection. | 0 |
| [1:0] | Reserved | RO | Reserved | 00b |

**FlashROM Status Register (R16_FLASH_STATUS)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:11] | Reserved | RO | Reserved | 00000b |
| [10:9] | Reserved | RO | Reserved | 01b |
| 8 | RB_ROM_READ_FREE | RO | FlashROM code and data area protection status:<br>1: The external programmer is readable;<br>0: Protected, externally inaccessible, and the program is kept secret. | 0 |

| | | | | |
|---|---|---|---|---|
| 7 | Reserved | RO | Reserved | 0 |
| 6 | RB_ROM_ADDR_OK | RO | FlashROM erase/program operation address valid flag It can be confirmed before or after the operation:<br>1: The address is valid;<br>0: The address is invalid. | 0 |
| [5:2] | Reserved | RO | Reserved | 0000b |
| 1 | RB_ROM_CMD_ERR | RO | FlashROM command response error:<br>1: Unknown command;<br>0: Command accepted. | 0 |
| 0 | RB_ROM_CMD_TOUT | RO | FlashROM operation results:<br>1: Timeout; 0: Successful. | 0 |

## 4.4 Flash-ROM Operation Procedure

1. Erase the Flash-ROM and change all data bits in the target sector to 1:

(1) Set the R8_FLASH_PROTECT register; turn on the erase/program to enable RB_ROM_DATA_WE or RB_ROM_CODE_WE bit, corresponding to DataFlash or CodeFlash. InfoFlash requires RB_ROM_DATA_WE and RB_ROM_CODE_WE both to be enabled;
(2) Set the address register R32_FLASH_ADDR, write in 32-bit address (the low 8 bits are invalid), and erase the entire block 512 bytes at a time;
(3) Set the command register R8_FLASH_COMMAND, write in 0A6H, and perform the operation of sector erase. The MCU automatically suspends running during the operation;
(4) After the operation is completed, the program resumes running. At this time, the status register R16_FLASH_STATUS is queried to check the operation status. If multiple sectors are to be erased, Step (2) to Step (4) are performed in cycles;
(5) Re-set the R8_FLASH_PROTECT register and turn off the erase/program enable control bit (RB_ROM_DATA_WE=0 or RB_ROM_CODE_WE=0).

2. Write the Flash-ROM to change part of the data bits in the target word from 1 to 0 (the bit data cannot be changed from 0 to 1):

(1) Set the R8_FLASH_PROTECT register; turn on the erase/program to enable RB_ROM_DATA_WE or RB_ROM_CODE_WE bit, corresponding to DataFlash or CodeFlash. InfoFlash requires RB_ROM_DATA_WE and RB_ROM_CODE_WE both to be enabled;
(2) Set the address register R32_FLASH_ADDR, write in the 32-bit address, require 4 bytes alignment;
(3) Set the data register R32_FLASH_DATA to be 4 bytes of data to be written;
(4) Set the command register R8_FLASH_COMMAND, write 09AH, and execute the operation of program/write-in. The MCU automatically suspends running during the operation;
(5) After the operation is completed, the program resumes running. At this time, the status register R16_FLASH_STATUS is queried to view the operation status. If multiple data is to be written, Step (2) to Step (4) are performed in cycles;
(6) Reset the R8_FLASH_PROTECT register to turn off the erase/program enable control bit (RB_ROM_DATA_WE=0 or RB_ROM_CODE_WE=0).

3. Read Flash-ROM:

Read the code or data of the target address by a pointer to the program memory space.

## 4.5 Unique Chip ID Number

Each chip is shipped with a unique ID number, namely the chip ID number. The ID data and its checksum are 8 bytes in total and are stored in the read-only area of the chip. For details, please refer to the example program.

# Chapter 5 Power Control

## 5.1 Power management

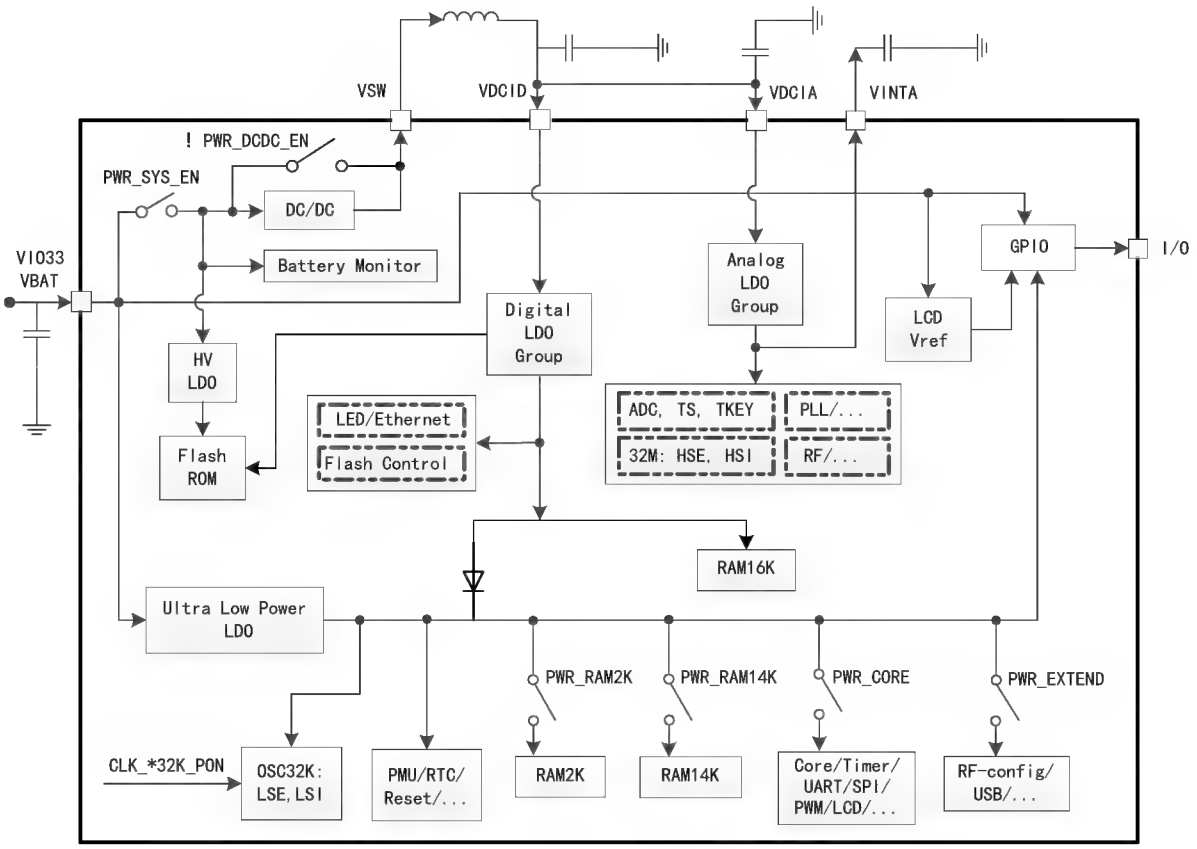
The CH579 has a built-in power management unit (PMU). The system power is input from VIO33 and provides the required power for the system's FlashROM, the system's digital circuits (including core, USB, Ethernet, LED, etc.) and the system's analog circuits (including high-frequency oscillators, PLLs, ADCs, RF transceivers, etc.) through the multiple built-in LDO voltage regulators.

There are two types of power supply during normal operation: direct power and DC-DC conversion. In addition to normal operation, the CH579 offers four types of low-power modes: idle mode, halt mode, sleep mode, and shutdown mode.

Instead of enabling DC-DC by default after power-on, a straight-through power supply is provided with a small voltage ripple. In order to reduce the system power consumption during normal operation, DC-DC can be enabled to increase the power consumption rate of the power supply, and the operating current will usually drop to about 60% of that of the straight-through mode.

In order to reduce the system power consumption during sleep, you can choose to turn off the system main LDO and switch to the auxiliary power supply provided by the system built-in ultra-low power ULP-LDO. When the system enters the mode of sleep or shutdown, in addition to the normal power supply units such as power management and RTC registers, you can choose whether to maintain power supply of the system high 2KB and 14KB SRAM, core kernel and all peripherals and whether to enable LSE/LSI.

Figure 5-1 Power system




## 5.2 Register description

Table 5-1 List of registers related to power management

| Name | Access Address | Description | Reset value |
|---|---|---|---|
| R16_SLP_CLK_OFF | 0x4000100C | Sleep clock control register | 0x0000 |
| R8_SLP_CLK_OFF0 | 0x4000100C | Sleep clock control register 0 | 0x00 |
| R8_SLP_CLK_OFF1 | 0x4000100D | Sleep clock control register 1 | 0x00 |

| | | | |
|---|---|---|---|
| R8_SLP_WAKE_CTRL | 0x4000100E | Wakeup event configuration register | 0x20 |
| R8_SLP_POWER_CTRL | 0x4000100F | Peripheral sleep power control register | 0x08 |
| R16_POWER_PLAN | 0x40001020 | Sleep power management register | 0x01DF |
| R16_AUX_POWER_ADJ | 0x40001022 | Auxiliary power adjustment control register | 0xXXXX |
| R8_BAT_DET_CTRL | 0x40001024 | Battery voltage detection control register | 0x00 |
| R8_BAT_DET_CFG | 0x40001025 | Battery voltage detection configuration register | 0x01 |
| R8_BAT_STATUS | 0x40001026 | Battery status register | 0x00 |

**Sleep Clock Control Register 0 (R8_SLP_CLK_OFF0)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | RB_SLP_CLK_UART3 | RWA | UART3 clock source:<br>1: Off; 0: On. | 0 |
| 6 | RB_SLP_CLK_UART2 | RWA | UART2 clock source:<br>1: Off; 0: On. | 0 |
| 5 | RB_SLP_CLK_UART1 | RWA | UART1 clock source:<br>1: Off; 0: On. | 0 |
| 4 | RB_SLP_CLK_UART0 | RWA | UART0 clock source:<br>1: Off; 0: On. | 0 |
| 3 | RB_SLP_CLK_TMR3 | RWA | Timer 3 clock source:<br>1: Off; 0: On. | 0 |
| 2 | RB_SLP_CLK_TMR2 | RWA | Timer 2 clock source:<br>1: Off; 0: On. | 0 |
| 1 | RB_SLP_CLK_TMR1 | RWA | Timer 1 clock source:<br>1: Off; 0: On. | 0 |
| 0 | RB_SLP_CLK_TMR0 | RWA | Timer 0 clock source:<br>1: Off; 0: On. | 0 |

**Sleep Clock Control Register 1 (R8_SLP_CLK_OFF1)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | RB_SLP_CLK_BLE | RWA | BLE controller clock source:<br>1: Off; 0: On. | 0 |
| 6 | RB_SLP_CLK_LED | RWA | LED controller clock source:<br>1: Off; 0: On. | 0 |
| 5 | RB_SLP_CLK_ETH | RWA | ETH controller clock source:<br>1: Off; 0: On. | 0 |
| 4 | RB_SLP_CLK_USB | RWA | USB controller clock source:<br>1: Off; 0: On. | 0 |
| 3 | RB_SLP_CLK_LCD | RWA | LCD controller clock source:<br>1: Off; 0: On. | 0 |
| 2 | RB_SLP_CLK_PWMX | RWA | PWMx clock source:<br>1: Off; 0: On. | 0 |
| 1 | RB_SLP_CLK_SPI1 | RWA | SPI1 clock source:<br>1: Off; 0: On. | 0 |
| 0 | RB_SLP_CLK_SPI0 | RWA | SPI0 clock source:<br>1: Off; 0: On. | 0 |

**Wakeup Event Configuration Register (R8_SLP_WAKE_CTRL)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7:6] | Reserved | RO | Reserved | 00b |
| 5 | RB_SLP_BAT_WAKE | RWA | Enable battery low voltage event wake-up system:<br>1: Enabled; 0: Disable. | 1 |
| 4 | RB_SLP_GPIO_WAKE | RWA | Enable the GPIO event wakeup system:<br>1: Enabled; 0: Disable. | 0 |
| 3 | RB_SLP_RTC_WAKE | RWA | Enable the RTC event wakeup system: | 0 |

| | | | | |
|---|---|---|---|---|
| | | | 1: Enabled; 0: Disable. | |
| 2 | Reserved | RO | Reserved | 0 |
| 1 | RB_SLP_ETH_WAKE | RWA | Enable the Ethernet event wakeup system:<br>1: Enabled; 0: Off. | 0 |
| 0 | RB_SLP_USB_WAKE | RWA | Enable the USB event wakeup system:<br>1: Enabled; 0: Off. | 0 |

**Peripheral Sleep Power Control Register (R8_SLP_POWER_CTRL)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7:6] | Reserved | RO | Reserved | 00b |
| 5 | RB_SLP_CLK_RAM2K | RWA | SRAM clock control of RAM2K:<br>1: Off; 0: On. | 0 |
| 4 | RB_SLP_CLK_RAMX | RWA | Clock control of the main SRAM (RAM16K + RAM14K)<br>1: Off; 0: On. | 0 |
| 3 | RB_SLP_ROM_PWR_DN | RWA | In halt mode, the FlashROM shutdown enable:<br>1: Put the ROM into the disabled state in halt mode;<br>0: The ROM is kept in the standby mode in halt mode. | 1 |
| 2 | Reserved | RO | Reserved | 0 |
| 1 | RB_SLP_ETH_PWR_DN | RWA | Ethernet transceiver ETH PHY power down enable:<br>1: Power down/disable;<br>0: Keep power supply. | 0 |
| 0 | Reserved | RO | Reserved | 0 |

**Sleep Power Management Register (R16_POWER_PLAN)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 15 | RB_PWR_PLAN_EN | RWA/ WZ | Sleep power planning control enable:<br>1: Start planning;<br>0: Close or end the plan. Turn on the power plan for execution when it enters the sleep or shutdown mode later. This bit is automatically cleared to zero after execution. | 0 |
| [14:11] | RB_PWR_MUST_0010 | RWA | Reserved. Must write 0010b. | 0000b |
| 10 | RB_PWR_DCDC_PRE | RWA | DC-DC bias circuit enable (immediately effective):<br>1: Enable; 0: Disabled. | 0 |
| 9 | RB_PWR_DCDC_EN | RWA | DC-DC enable bit (immediately effective):<br>1: DC-DC is enabled, and the pass-through power is off;<br>0: DC-DC is disabled and the pass-through power is on. | 0 |
| 8 | RB_PWR_LDO_EN | RWA | Internal LDO control (sleep planning)<br>1: Turn on LDO;<br>0: Plan to turn off LDO, saving more power. | 1 |
| 7 | RB_PWR_SYS_EN | RWA | System power control (sleep planning):<br>1: Provide system power (on the VSW pin);<br>0: Turn off the system power, and plan to enter the sleep mode or power-off mode. | 1 |
| 6 | Reserved | RWA | Reserved. Must write 0. | 1 |
| 5 | Reserved | RO | Reserved | 0 |
| 4 | RB_PWR_RAM14K | RWA | SRAM power supply of RAM14K (sleep planning): | 1 |

| | | | | |
|---|---|---|---|---|
| | | | 1: Dual power supply;<br>0: No auxiliary power supply required. | |
| 3 | RB_PWR_EXTEND | RWA | USB and RF configuration power supply (sleep planning):<br>1: Dual power supply;<br>0: No auxiliary power supply. | 1 |
| 2 | RB_PWR_CORE | RWA | Power supply of the core and basic peripherals (sleep planning):<br>1: Dual power supply;<br>0: No auxiliary power supply. | 1 |
| 1 | RB_PWR_RAM2K | RWA | SRAM power supply of RAM2K (sleep planning):<br>1: Dual power supply;<br>0: No auxiliary power supply. | 1 |
| 0 | Reserved | RO | Reserved | 1 |

Bits of this register other than RB_PWR_DCDC_PRE and RB_PWR_DCDC_EN are all sleep planning preset, and its power configuration takes effect after entering the low-power sleep mode and the shutdown mode.

**Auxiliary Power Adjustment Control Register (R16_AUX_POWER_ADJ)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:8] | Reserved | RO | Reserved, read-only bits, write operations are not affected. | XXb |
| [7:6] | Reserved | RO | Reserved | 00b |
| [5:3] | Reserved | RWA | Reserved. The original value must be kept unchanged when write-in. | 1XXb |
| [2:0] | RB_ULPLDO_ADJ | RWA | Auxiliary power supply output voltage regulation value of ultra-low power LDO (The values are only for reference and not recommended to modify.):<br>000: 0.908V; 001: 0.931V;<br>010: 0.954V; 011: 0.977V;<br>100: 1.000V; 101: 1.023V;<br>110: 1.046V; 111: 1.069V. | XXXb<br>(100b) |

**Battery Voltage Detection Control Register (R8_BAT_DET_CTRL)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7:4] | Reserved | RO | Reserved | 0000b |
| 3 | RB_BAT_LOW_IE | RWA | Battery low voltage interrupt enable:<br>1: Enabled; 0: Off. | 0 |
| 2 | RB_BAT_LOWER_IE | RWA | Battery ultra-low voltage interrupt enable:<br>1: Enabled; 0: Off. | 0 |
| 1 | Reserved | RO | Reserved | 0 |
| 0 | RB_BAT_DET_EN | RWA | Battery voltage detection feature enable:<br>1: Enable low-voltage detection, and turn on the reference voltage and other modules at the same time. The current is 210uA in sleep mode;<br>0: Off. | 0 |

*Note: If the battery voltage reaches the detection threshold value of ultra-low voltage and both RB_BAT_LOWER_IE and RB_BAT_LOW_IE are enabled (normally only one of them is enabled), non- maskable interrupt (NMI) will be generated, which is equivalent to increasing the interrupt priority.*

**Battery Voltage Detection Configuration Register (R8_BAT_DET_CFG)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7:2] | Reserved | RO | Reserved | 000000b |
| [1:0] | RB_BAT_LOW_VTH | RWA | Set the detection threshold for low voltage | 01b |

| | | | | |
|---|---|---|---|---|
| | | | and ultra-low voltage:<br>(Ultra-low voltage reference threshold, low voltage reference threshold)<br>00:1.97V , 2.25V;<br>01:2.05V , 2.33V;<br>10:2.13V , 2.41V;<br>11:2.21V , 2.49V. | |

**Battery Status Register (R8_BAT_STATUS)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7:2] | Reserved | RO | Reserved | 000000b |
| 1 | RB_BAT_STAT_LOW | RO | The battery is in a low voltage state:<br>1: Below the low voltage threshold;<br>0: No. | 0 |
| 0 | RB_BAT_STAT_LOWER | RO | The battery is in an ultra-low voltage state:<br>1: Below the ultra-low voltage threshold;<br>0: No. | 0 |

## 5.3 Low-power mode

After the system is reset, the micro-controller is in normal run status. When the MCU is not required to run, the appropriate low power mode can be selected to save power. The user needs to select a suitable low power mode based on conditions such as the lowest power consumption, the fastest start time, and available wake-up events.
A chip is provided with the following four main low-power modes:

- Idle mode

All peripherals remain powered; the core stops running; the clock system runs. Once a wake-up event is detected, it can be woken up immediately.

- Halt mode

On the basis of the idle mode, the clock system stops, and the FlashROM can be put into the stop mode to further reduce power consumption. After the wakeup event is detected, the clock runs firstly and then the kernel is woken up to run.

- Sleep mode

The main LDO is turned off. The ultra-low-power ULP-LDO maintains the PMU, the core, and the basic peripherals. Users can choose whether to turn on LSE or LSI and whether to maintain the power supply of RAM2K, RAM14K, USB, and RF configurations. Upon detection of a wake-up event, the main LDO is first turned on; then the clock runs; in the end, the core is woken up and the program continues to run. A higher frequency can be reset when needed.

- Shutdown mode:

Based on the sleep mode, the core, basic peripherals and USB and RF configurations are turned off. Users can choose whether to turn on the LSE or LSI and whether to maintain the power supply of RAM2K and RAM14K. Upon detection of a wake-up event, the PMU will perform the GRWSM reset and the software can distinguish RPOR based on the reset flag of RB_RESET_FLAG and the data retention area in the optional RAM.

The following table details the features and wake-up paths of several low-power modes:

Table 5-2 Low-power modes

| Mode | Feature | Access conditions | Wake-up event | Power consumption |
|---|---|---|---|---|
| Idle | The peripherals are powered normally; the core stops running; the clock system is in operation. Users can select to turn off the peripheral clocks by the peripheral clock control bits. | Set the core control bit SCR=0, set the wake-up conditions, and then execute_WFI () or _WFE (). | I/O or RTC or BAT or USB or ETH | 1.15mA~1.5mA |

| | | | | |
|---|---|---|---|---|
| Halt | The peripherals are powered normally; the core stops running; the clock system stops (PLL/HSE/HSI stops). You can select FlashROM to standby or stop. | Set the core control bit SCR=1, set the wake-up conditions, and then execute _WFI () or _WFE (). | I/O or RTC or BAT or USB or ETH | 420uA~470uA |
| Sleep | The main LDO is off. The ultra-low-power ULP-LDO maintains the power supply of the PMU, the core, and basic peripherals. Users can select whether to turn on or off LSE or LSI and whether to maintain the power supply of RAM2K, RAM14K, USB, and RF configurations. | Set the core control bit SCR=1, set POWER_PLAN, set the wake-up conditions, and then execute WFI() or WFE(). | I/O or RTC or BAT. The chip will continue to run after wakeup. | 0.6uA~2.6uA |
| Shutdown | The ultra-low-power LDO maintains the power supply of the PMU. Users can select whether to turn on LSE or LSI and whether to maintain the power supply of RAM2K and RAM14K for data retention. | Set the core control bit SCR=1, set POWER_PLAN, set the wake-up conditions, and then execute WFI() or WFE(). | I/O or RTC or BAT. The chip will automatically reset after wakeup. | 0.2uA~1.5uA |

The following table describes the detailed configurations of several low-power modes:

Table 5-3 Example of detailed configurations of low-power modes

| Planning configurtion | SYS_EN | RAM2K | RAM14K | CK32K | CORE | EXTEND | Power consumption (for reference only) |
|---|---|---|---|---|---|---|---|
| Function to maintain power supply | System power supply VSW | Data area 2KB | Data area 14KB | Either LSE or LSI, RTC wake-up | MO core and basic peripherals | USB and RF configurations | PMU and RTC registers are always powered, about 0.2uA |
| Common configurations of shutdown mode | 0 | 0 | 0 | 0 | 0 | 0 | 0.2uA |
| | 0 | 1 | 0 | 0 | 0 | 0 | 0.5uA~0.7uA |
| | 0 | 0 | 0 | 1 | 0 | 0 | 0.5uA~0.6uA |
| | 0 | 1 | 0 | 1 | 0 | 0 | 0.8uA~1.1uA |
| Common configurations of sleep mode | 0 | 1 | 0 | 0 | 1 | 0 | 0.6uA~0.8uA |
| | 0 | 1 | 0 | 1 | 1 | 0 | 0.9uA~1.2uA |
| | 0 | 0 | 1 | 0 | 1 | 0 | 1.1uA~1.5uA |
| | 0 | 0 | 1 | 1 | 1 | 0 | 1.4uA~1.9uA |
| | 0 | 1 | 1 | 1 | 1 | 1 | 1.8uA~2.6uA |

## 5.4 DC-DC operation procedures

Enable DC-DC power mode (Confirm the inductance and capacitance required by DC-DC on the external hardware circuit before enabling):

(1) Enter safe access mode: Register R8_SAFE_ACCESS_SIG first writes 0x57 and then writes 0xA8;
(2) Open the DC-DC bias circuit: Set RB_PWR_DCDC_PRE of Register R16_POWER_PLAN to 1;
(3) Delay at least 10us;
(4) Enter the safe access mode again: first write 0x57 into the register R8_SAFE_ACCESS_SIG and then write 0xA8;
(5) Enable DC-DC power: Set RB_PWR_DCDC_EN of Register R16_POWER_PLAN to 1 to enable DC-DC.

Turn off DC-DC and switch to the pass-through mode:

(1) Enter the safe access mode: first write 0x57 into the register R8_SAFE_ACCESS_SIG and then write 0xA8;
(2) Reset the control bits of RB_PWR_DCDC_EN and RB_PWR_DCDC_PRE of Register R16_POWER_PLAN to zero.

# Chapter 6 System Clock and RTC

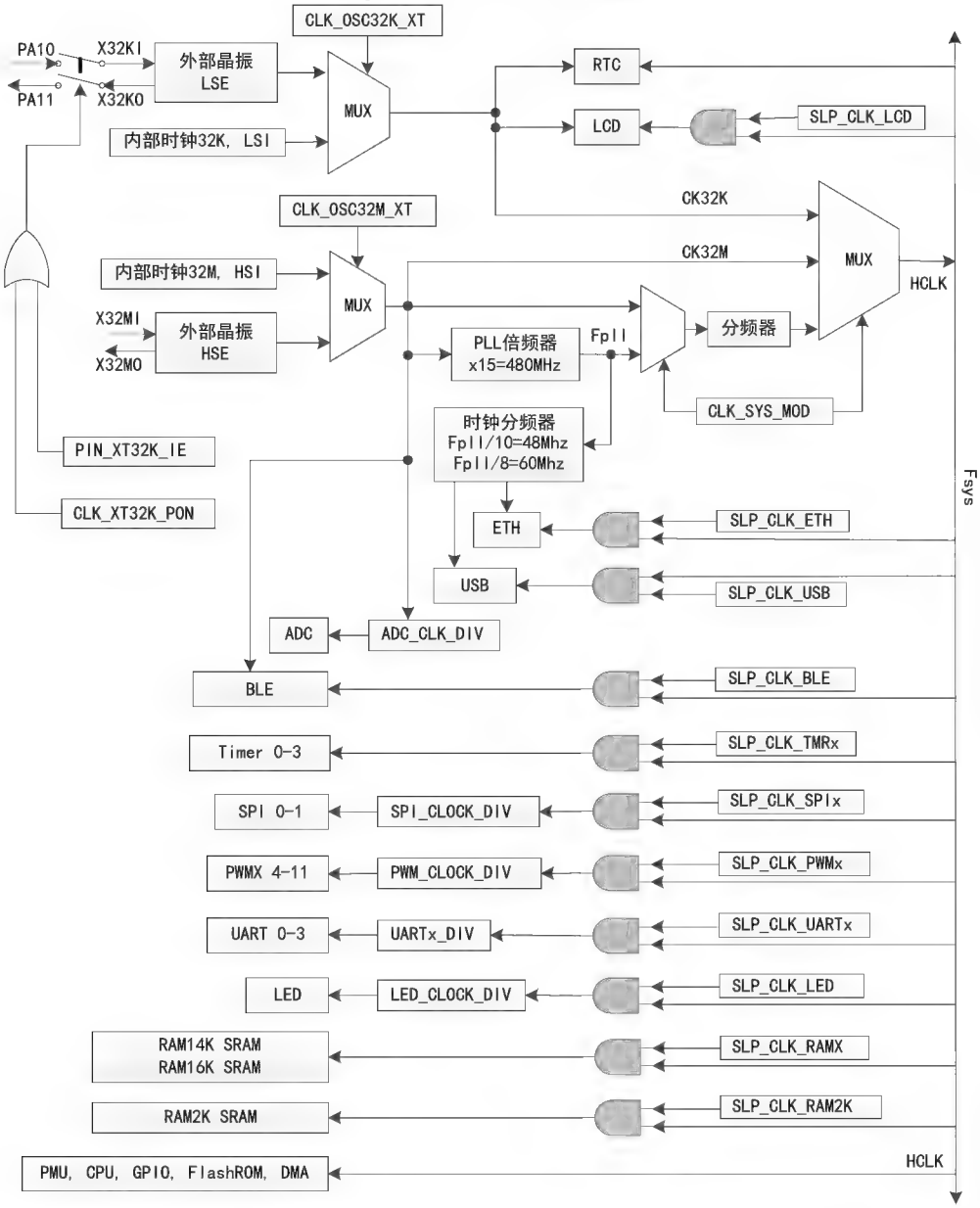
## 6.1 System clock introduction

The following different clock sources can be selected to drive the system clock HCLK (Fsys)
- HSE or HSI original clock CK32M.
- Frequency division of HSE or HSI.
- Frequency division of internal PLL (480MHz at default).
- LSE or LSI original clock CK32K.

Any clock source can be independently turned on or off, thereby optimizing the power consumption of the system.

### 6.1.1 Clock structure

Figure 6-1 Clock tree block




The figure above indicates the clock tree structure of the internal system. Specifically, the features of RTC and LCD use the 32KHz clock source CK32K, so the low frequency clock must be turned on when these features are used; the data transmission of USB and Ethernet depends on the clock source generated by the PLL frequency division; other peripheral drive clocks and digital control logic are driven by the system clock or by frequency re-division.

## 6.2 RTC instruction

Real Time Clock (RTC) is an independent timer that contains a set of counters counting continuously. The simple feature of calendar is available under the appropriate software configurations. The value of the counter is modified to reset the current time and date.
The RTC registers are powered as often as the PMU, and the RTC settings and time remain unchanged after a system reset or a wake-up from a low-power mode.

### 6.2.1 Main features

- 2 modes configurable:
  - Timing mode: The software can select a fixed cycle time (timing) to generate an interrupt notification.
  - Trigger mode: Match a software-preset target alarm time and generate an interrupt notification.
- 3 groups of 16-bit counters that provide the count of the CK32K original cycle, the 2 second cycle, and the 1-day cycle.

## 6.3 Register descriptions

Table 6-1 List of registers related to clock and oscillator control

| Name | Access address | Description | Reset value |
|---|---|---|---|
| R16_CLK_SYS_CFG | 0x40001008 | System clock configuration register | 0x05 |
| R8_HFCK_PWR_CTRL | 0x4000100A | High frequency clock module power control register | 0x0C |
| R16_INT32K_TUNE | 0x4000102C | Internal 32KHz clock calibration register | 0x0200 |
| R8_XT32K_TUNE | 0x4000102E | External 32KHz clock resonance control register | 0xC3 |
| R8_CK32K_CONFIG | 0x4000102F | 32KHz oscillator configuration register | 0xX2 |
| R8_INT32M_CALIB | 0x4000104C | Internal 32MHz clock calibration register | 0xXX |
| R8_XT32M_TUNE | 0x4000104E | External 32MHz clock resonance control register | 0x31 |
| R16_OSC_CAL_CNT | 0x40001050 | Oscillator frequency calibration count value register | 0xXXXX |
| R8_OSC_CAL_CTRL | 0x40001052 | Oscillator frequency calibration control register | 0x02 |
| R8_PLL_CONFIG | 0x4000104B | PLL configuration register | 0xX0 |
| R8_RTC_FLAG_CTRL | 0x40001030 | RTC flag and control register | 0x30 |
| R8_RTC_MODE_CTRL | 0x40001031 | RTC mode configuration register | 0x02 |
| R32_RTC_TRIG | 0x40001034 | RTC trigger value register | 0x00000000 |
| R16_RTC_CNT_32K | 0x40001038 | RTC-based 32768Hz count value register | 0xXXXXXXXX |
| R16_RTC_CNT_2S | 0x4000103A | RTC count value register in the unit of 2s | 0xXXXXXXXX |
| R32_RTC_CNT_DAY | 0x4000103C | RTC count value register in the unit of day | 0x0000XXXX |

**System Clock Configuration Register (R16_CLK_SYS_CFG)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 15 | RB_XO_DI | RO | Input state sample value of the X32MO pin. | 0 |
| [14:10] | Reserved | RO | Reserved | 00000b |
| 9 | RB_CLK_OSC32M_XT | RWA | CK32M(32MHz) clock source selection bit:<br>1: External 32MHz oscillator;<br>0: Internal 32MHz oscillator. | 0 |
| 8 | Reserved | RO | Reserved | 0 |
| [7:6] | RB_CLK_SYS_MOD | RWA | HCLK system clock source mode selection:<br>00: CK32M (default 32MHz) for frequency division;<br>01: PLL (default 480MHz) for frequency division;<br>10: CK32M (default 32MHz) as HCLK;<br>11: CK32K (default 32KHz) as HCLK. | 00b |
| 5 | Reserved | RO | Reserved | 0 |
| [4:0] | RB_CLK_PLL_DIV | RWA | HCLK output clock division factor, the minimum value is 2, 0 represents the maximum value 32. Write 1 to turn off HCLK. | 00101b |

Calculations:
Fck32m = RB_CLK_OSC32M_XT ? XT_32MHz : RC_32MHz;

Fck32k = RB_CLK_OSC32K_XT ? XT_32KHz : RC_32KHz;
Fpll = Fck32m * 15 = 480MHz;
Fsys = RB_CLK_SYS_MOD[1] ? ( RB_CLK_SYS_MOD[0] ? Fck32k : Fck32m ) :
( ( RB_CLK_SYS_MOD[0] ? Fpll : Fck32m ) / RB_CLK_PLL_DIV );
Default power-on value Fsys = Fck32m / RB_CLK_PLL_DIV = 32MHz / 5 = 6.4MHz;
Fsys range: 32KHz, 1MHz~16MHz, 32MHz, 15MHz~40MHz

**High Frequency Clock Module Power Control Register (R8_HFCK_PWR_CTRL)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7:6] | Reserved | RWA | Reserved. Must write 00. | 00b |
| 5 | Reserved | RO | Reserved | 0 |
| 4 | RB_CLK_PLL_PON | RWA | PLL power control bit:<br>1: Power on; 0: Power down. | 0 |
| 3 | RB_CLK_INT32M_PON | RWA | Internal 32MHz oscillator power control bit:<br>1: Power on; 0: Power down. | 1 |
| 2 | RB_CLK_XT32M_PON | RWA | External 32MHz oscillator power control bit:<br>1: Power on; 0: Power down. | 1 |
| [1:0] | Reserved | RO | Reserved | 00b |

**Internal 32KHz Clock Calibration Register (R16_INT32K_TUNE)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:10] | Reserved | RO | Reserved | 000000b |
| [9:0] | RB_INT32K_TUNE | RWA | The calibration value of internal RC 32KHz clock frequency. | 1000000000b |

**External 32KHz Clock Resonance Control Register (R8_XT32K_TUNE)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7:4] | RB_XT32K_C_LOAD | RWA | Choose a built-in load capacitor that matches the external 32KHz crystal (which may affect the accuracy of RTC clock):<br>0000: Built-in load capacity is 2pF;<br>Other:<br>Capacitance = RB_XT32K_C_LOAD + 12pF.<br>0001b~1111b correspond to approximately 13pF to 27pF, respectively. Choose according to the crystal parameters used. | 1100b |
| [3:2] | Reserved | RO | Reserved | 00b |
| [1:0] | RB_XT32K_I_TUNE | RWA | Select the bias current of the external 32KHz oscillator:<br>00: 70% of the rated current;<br>01: Rated current;<br>10: 140% of the rated current;<br>11: 200% of the rated current. | 11b |

**32KHz Oscillator Configuration Register (R8_CK32K_CONFIG)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | RB_32K_CLK_PIN | RO | 32KHz clock pin status (asynchronous signal) | X |
| [6:3] | Reserved | RO | Reserved | 0000b |
| 2 | RB_CLK_OSC32K_XT | RWA | CK32K (32KHz) clock source selection bit:<br>1: External 32KHz oscillator;<br>0: Internal 32KHz oscillator. | 0 |
| 1 | RB_CLK_INT32K_PON | RWA | Internal 32KHz oscillator power control bit:<br>1: Power on; 0: Power down. | 1 |
| 0 | RB_CLK_XT32K_PON | RWA | External 32KHz oscillator power control bit:<br>1: Power on; 0: Power down. | 0 |

**Internal 32MHz Clock Calibration Register (R8_INT32M_CALIB)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7:0] | R8_INT32M_CALIB | RWA | Frequency calibration value of the internal RC 32MHz clock. | XXXXXXXXb |

**External 32MHz Clock Resonance Control Register (R8_XT32M_TUNE)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | Reserved | RO | Reserved | 0 |
| [6:4] | RB_XT32M_C_LOAD | RWA | Select a built-in load capacitor that matches the external 32MHz crystal (possibly affecting wireless communication):<br>Capacitance = RB_XT32M_C_LOAD* 2 + 10pF. 000b~111b correspond to approximately 10pF ~ 24pF respectively.<br>Depending on the selected parameters of the crystal used, the commonly used value is 111b. | 011b |
| [3:2] | Reserved | RO | Reserved | 00b |
| [1:0] | RB_XT32M_I_BIAS | RWA | Select the bias current of the external 32MHz oscillator:<br>00: 75% of the rated current;<br>01: Rated current;<br>10: 125% of the rated current;<br>11: 150% of the rated current. | 01b |

**Oscillator Frequency Calibration Counter Value Register (R16_OSC_CAL_CNT)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:12] | Reserved | RO | Reserved | 0000b |
| [11:0] | RB_OSC_CAL_CNT | RO | The count value of five CK32K cycles based on the system main frequency and are used to calibrate the frequency of the internal 32KHz oscillator. | XXXh |

**Oscillator Frequency Calibration Control Register (R8_OSC_CAL_CTRL)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7:2] | Reserved | RO | Reserved | 000000b |
| 1 | RB_OSC_CNT_HALT | RO | Count status bit of oscillator frequency calibration counter:<br>1: Counting is being paused;<br>0: Counting is in progress. | 1 |
| 0 | RB_OSC_CNT_EN | RWA | Oscillator frequency calibration counter enable bit:<br>1: Enable counting; 0: Disable counting. | 0 |

**PLL Configuration Register (R8_PLL_CONFIG)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | RB_PLL_LOCKED | RO | PLL lock status:<br>1: Locking; 0: Not locked currently. | X |
| [6:2] | Reserved | RO | Reserved | 00000b |
| [1:0] | RB_PLL_CFG_DAT | RWA | PLL configuration parameters. | 00b |

**RTC Flag and Control Register (R8_RTC_FLAG_CTRL)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | RB_RTC_TRIG_FLAG | RO | Activation flag of RTC trigger mode. | 0 |
| 6 | RB_RTC_TMR_FLAG | RO | Activation flag of RTC timing mode. | 0 |
| 5 | RB_RTC_TRIG_CLR | RW | When the trigger mode is disabled, this bit is | 1 |

| | | | | |
|---|---|---|---|---|
| | | | fixed as 1. When the trigger mode is enabled, write 1, clear the trigger mode activation flag RB_RTC_ TRIG FLAG, and automatically clear to 0. | |
| 4 | RB_RTC_TMR_CLR | RW | When the timing mode is disabled, this bit is fixed as 1. When the timing mode is enabled, write 1, clear the timing mode activation flag RB_RTC_TMR_FLAG, and automatically clear to 0. | 1 |
| [3:0] | Reserved | RO | Reserved | 0000b |

**RTC Mode Configuration Register (R8_RTC_MODE_CTRL)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | RB_RTC_LOAD_HI | RWA | Write 1 to load the high words of the RTC counter and automatically reset to zero after loading. Load R32_RTC_TRIG (actually only the lower 14 bits) into R32_RTC_CNT_DAY. | 0 |
| 6 | RB_RTC_LOAD_LO | RWA | Write 1 to load the low words of the RTC counter and automatically reset to zero after loading. Load the high 16 bits of R32_RTC_TRIG into R16_RTC_CNT_2S; load the low 16 bits of R32_RTC_TRIG into R16_RTC_CNT_32K. | 0 |
| 5 | RB_RTC_TRIG_EN | RWA | RTC Trigger Mode Enable Bit:<br>1: Enabled; 0: Disabled. | 0 |
| 4 | RB_RTC_TMR_EN | RWA | RTC Timing Mode Enable Bit:<br>1: Enabled; 0: Disabled. | 0 |
| 3 | RB_RTC_IGNORE_B0 | RWA | Compare/ignore the lowest bit of the match value in trigger mode:<br>1: ignore the lowest bit; 0: compare the lowest bit. | 0 |
| [2:0] | RB_RTC_TMR_MODE | RWA | Select the fixed cycle (timing) of RTC timing mode:<br>000: 0.125S; 001: 0.25S;<br>010: 0.5S; 011: 1S;<br>100: 2S; 101: 4S;<br>110: 8S; 111: 16S. | 010b |

**RTC Trigger Value Register (R32_RTC_TRIG)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:0] | R32_RTC_TRIG | RWA | Preset match values in the RTC trigger mode. The higher 16 bits and lower 16 bits match R16_RTC_CNT_2S and R16_RTC_CNT_32K, respectively.<br>Match with RB_RTC_LOAD_LO and RB_RTC_LOAD_HI to update the current value of the RTC counter. | 0000h |

*Note: The preset match values are not directly written into the target time. For simple calculations, please refer to the description below.*

**RTC-based 32768Hz Count Value Register (R16_RTC_CNT_32K)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:0] | R16_RTC_CNT_32K | RO | RTC-based 32768Hz count value register. | XXXXh |

**RTC Count Value Register in the Unit of 2S (R16_RTC_CNT_2S)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:0] | R16_RTC_CNT_2S | RO | The current count value of RTC in the unit of 2S. | XXXXh |

**RTC Count Value Register in the Unit of days (R32_RTC_CNT_DAY)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:16] | Reserved | RO | Reserved | 0000h |
| [15:14] | Reserved | RO | Reserved | 00b |
| [13:0] | R32_RTC_CNT_DAY | RO | The current count value of RTC in the unit of day. | XXXXXXX XXXXXXXb |

## 6.4 Function description and configuration

### 6.4.1 RTC counter initialization

(1) Set the value of R32_RTC_TRIG register, and set RB_RTC_LOAD_HI to load the value of R32_RTC_TRIG register into R32_RTC_CNT_DAY Register;
(2) Set the value of R32_RTC_TRIG register, and set RB_RTC_LOAD_LO to load the higher and lower 16-bit value of the R32_RTC_TRIG register to the R16_RTC_CNT_2S Register and the R16_RTC_CNT_32K Register respectively.

### 6.4.2 RTC clock source switched to LSE crystal

(1) Confirm that the GPIO pins where X32KI and X32KO are located are not set as output, that no pull-up and pull-down resistors are set, that only crystals are set;
(2) Configure R8_CK32K_CONFIG Register, set RB_CLK_XT32K_PON to 1, and turn on the external 32KHz crystal oscillator;
(3) It is recommended to set RB_XT32K_I_TUNE to the maximum, wait for the crystal oscillator to stabilize (about several hundreds of mS), and then change to the rated current;
(4) Configure the R8_CK32K_CONFIG register, set RB_CLK_OSC32K_XT to 1, and request to switch the clock source to the crystal oscillator;
(5) Wait for at least half of the 32KHz clock cycle, usually 16uS, to actually complete the switching of clock sources.

### 6.4.3 RTC timing function

(1) Configure the R8_RTC_MODE_CTRL register, set RB_RTC_TMR_MODE to select the appropriate timing cycle, set RB_RTC_TMR_EN to 1, and turn on the feature of RTC timing;
(2) After the timing cycle is reached, the RTC timing activation flag of RB_RTC_TMR_FLAG and interrupt will be generated. Inquiry the R8_RTC_FLAG_CTRL register and set the RB_RTC_TMR_CLR to clear flag.

### 6.4.4 RTC trigger function

(1) Set the target match values in the R32_RTC_TRIG Register. The steps of calculation and operation are: Calculate the target time value by adding the current time R32_RTC_CNT_32K(high 16 bits R16_RTC_CNT_2S and low 16 bits R16_RTC_CNT_32K) with DelayTime (in S), T32 = R32_RTC_CNT_32K + DelayTime * 32768,
If (T32 & 0xFFFF)! = 0, T32 = T32 + 0x10000,
Write T32 to the R32_RTC_TRIG register to complete the setting of match values.
(2) Configure the R8_RTC_MODE_CTRL register, set RB_RTC_TRIG_EN to 1, and turn on the feature of RTC trigger;
(3) When the current count values of RTC, R16_RTC_CNT_2S and R16_RTC_CNT_32K, match the preset high and low 16 bits of the R32_RTC_TRIG respectively, generate the RTC trigger activation flag RB_RTC_TRIG_FLAG and interrupt, and set the RB_RTC_TRIG_CLR to clear flag.
(4) If the RTC has been over-calibrated, the target absolute time trigger can be supported. The target time value T32 is calculated based on the target year/month/day/hour/minute/second/millisecond. The other steps are the same as above. For details, please refer to the example program of evaluation board.

### 6.4.5 Calibrate the internal 32K clock LSI with HSE

Please refer to the example program of evaluation board.

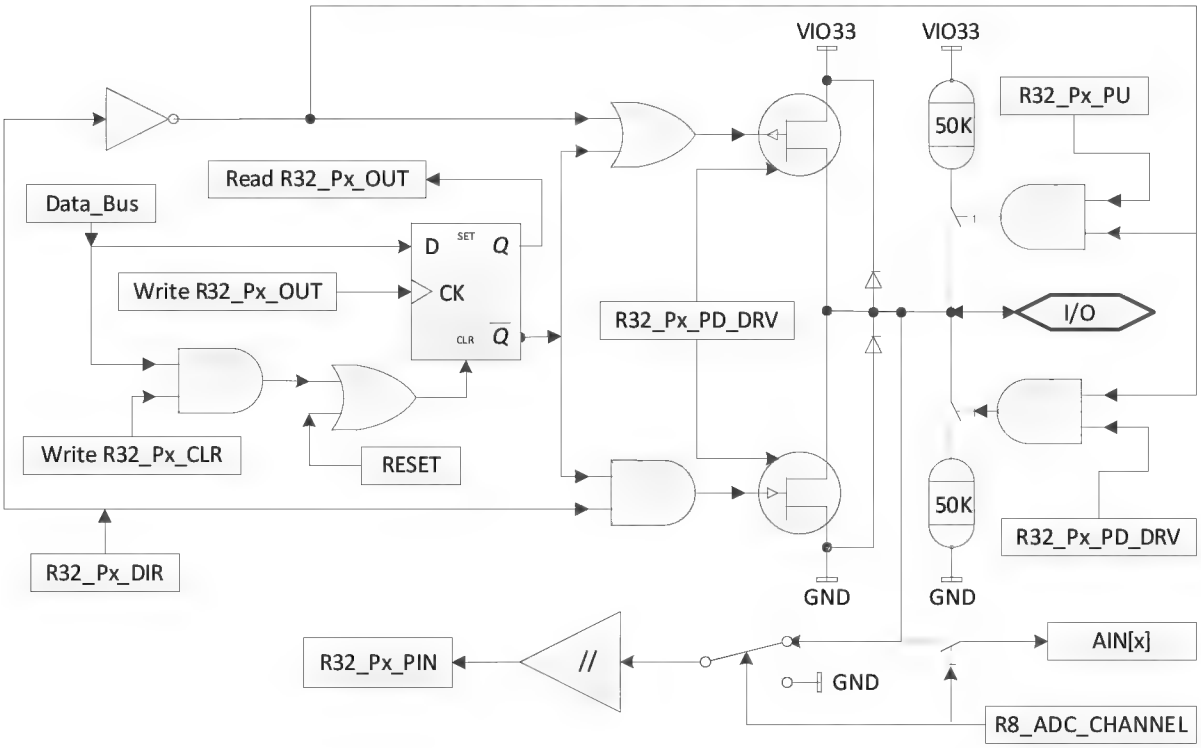
# Chapter 7 General-purpose I/O and Multiplexing Functions

## 7.1 GPIO introduction

The chip provides two sets of GPIO ports: PA and PB. There are a total of 40 general-purpose input and output pins, some of which have the features of interrupt, multiplexing and mapping.
Each GPIO port has a 32-bit direction configuration register R32_Px_DIR, a 32-bit pin input register R32_Px_PIN, a 32-bit data output register R32_Px_OUT, a 32-bit data reset register R32_Px_CLR, a 32-bit pull- up resistor configuration register R32_Px_PU and a 32-bit register R32_Px_PD_DRV with pull-down resistor/drive capability configurations.
In PA ports, PA[0]~PA[15] bits are valid, corresponding to 16 GPIO pins on the chip. In PB ports, PB[0]~PB[23] bits are valid, corresponding to 24 GPIO pins on the chip. Among them, 32 I/O pins, PA[0]~PA[15] and PB[0]~PB[15], have interrupt feature and can realize the feature of wakeup from sleep.
Each I/O port bit is freely programmable, but the I/O port registers must be accessed by 8-bit, 16-bit or 32-bit words. If the pin multiplexing feature is not enabled, it is used by default as a general purpose I/O port.

The following is the block diagram of the internal GPIO structure:

Figure 7-1 Block diagram of I/O internal structure




## 7.2 External interrupt/wakeup

32 I/O pins (PA[0]~PA[15], PB[0]~PB[15]) on chip have the feature of interrupt and can realize sleep wake-up.
In order to use an external interrupt, the port bits must be configured as the input mode. Four trigger modes are available: high level, low level, rising edge, and falling edge.
The wake-up feature requires the port bit interrupt to enable R16_Px_INT_EN and turn on the GPIO wake-up control bit RB_SLP_GPIO_WAKE in the register R8_SLP_WAKE_CTRL.

## 7.3 GPIO Multiplexing and Remapping

### 7.3.1 Multiplexing functions

Some I/O pins have the feature of multiplexing. By default, all I/O pins have the general-purpose I/O feature after power-on. After each feature module is enabled, the corresponding original GPIO pins are configured as feature pins corresponding to their respective feature module.

If a pin multiplexes multiple features that have been enabled, the priority order of the multiplexing feature can refer to the sequence of features as described in Section 1.2 of "Multiplexing".
For example, if the PA0 pin is multiplexed to SCK1/SLVA/LED0, the clock feature of SPI1 takes precedence, and the feature of serial data output 0 of the LED screen interface ranks the lowest. In this way, among multiple multiplexing features, pins that do not need to be used and have the lowest priority of feature are enabled with multiplexing feature at a relatively higher priority.
The following table lists some of the GPIO configurations used for the feature pins of the peripheral modules.

Table 7-1 Timer x

| TMR0/1/2/3 pin | Feature configuration | GPIO configuration |
|---|---|---|
| TMRx | Input capture channel x | Input (floating input / pull-up input / pull-down input) |
| | Output PWM channel x | Push-pull output |

Table 7-2 UARTx

| UART0/1/2/3 pin | Feature configuration | GPIO configuration |
|---|---|---|
| TXDx | UART transmitter x | Push-pull output |
| RXDx | UART receiver x | Pull-up input (recommended) or floating input |
| RTS，DTR | MODEM signal output or RS485 control | Push-pull output |
| CTS, DSR, RI, DCD | MODEM signal input | Pull-up input (recommended) or floating input |

Table 7-3 SPIx

| SPI0/1 pin | Feature configuration | GPIO configuration |
|---|---|---|
| SCKx | Clock output in master mode | Push-pull output |
| | Clock input in slave mode | Input (floating input / pull-up input / pull-down input) |
| MOSIx | Full duplex mode - master mode | Push-pull output |
| | Full duplex mode - slave mode | Input (floating input / pull-up input / pull-down input) |
| | Half-duplex mode - master mode | Not used. Can be used as general purpose I/O |
| | Half-duplex mode - slave mode | Not used. Can be used as general purpose I/O |
| MISOx | Full duplex mode - master mode | Input (floating input / pull-up input / pull-down input) |
| | Full-duplex mode - slave mode | Input (pull-up recommended; automatically switch to push-pull output after chip selection) or push-pull output (disabled for bus connection) |
| | Half-duplex mode - master mode | Input or push-pull output, manual switching |
| | Half-duplex mode - slave mode | Input (pull-up recommended; automatically switch to push-pull output after chip selection) |
| SCS | Chip selection output in master mode | Push-pull output (can be replaced with other pins) |
| | Chip selection input in slave mode | Pull-up input (recommended) or floating input |

Table 7-4 ADC

| ADC sampling channel pin | Feature configuration | GPIO configuration |
|---|---|---|
| AINx | Analog to digital conversion input channel | Floating input |

Table 7-5 USB

| USB signal pin | Feature configuration | GPIO configuration |
|---|---|---|
| USB_DM | Connect to internal USB transceiver | Floating input |

| | | |
|---|---|---|
| USB_DP | Connect to internal USB transceiver | Floating input |

Table 7-6 Ethernet

| Ethernet signal pin | Feature configuration | GPIO configuration |
|---|---|---|
| ET+/ET- | Connect to an internal Ethernet transceiver | Floating input |
| ER+/ER- | Connect to an internal Ethernet transceiver | Floating input |

### 7.3.2 Remapping of function pins

In order to enable the peripheral functions and optimize the utilization rate at the same time, some function pins can be remapped to other pins by setting the R16_PIN_ALTERNATE feature pin remapping register.

Table 7-7 Multiplexing function remapped to pins

| Peripheral function pins | Default GPIO pins | Remapped GPIO pins |
|---|---|---|
| SPI0 | PA[12]/PA[13]/PA[14]/PA[15] | PB[12]/PB[13]/PB[14]/PB[15] |
| RXD3/TXD3 | PA[4]/PA[5] | PB[20]/PB[21] |
| RXD2/TXD2 | PA[6]/PA[7] | PB[22]/PB[23] |
| RXD1/TXD1 | PA[8]/PA[9] | PB[8]/PB[9] |
| RXD0/TXD0 | PB[4]/PB[7] | PA[15]/PA[14] |
| TMR3/PWM3/CAP3 | PA[2] | PB[18] |
| TMR2/PWM2/CAP2 | PA[11] | PB[11] |
| TMR1/PWM1/CAP1 | PA[10] | PB[10] |
| TMR0/PWM0/CAP0 | PA[3] | PB[19] |

## 7.4 Register description

| Name | Access address | Description | Reset value |
|---|---|---|---|
| R16_PIN_ALTERNATE | 0x40001018 | Feature pin remapping register | 0x0000 |
| R16_PIN_ANALOG_IE | 0x4000101A | Peripheral analog pin configuration register | 0x0000 |
| R16_PA_INT_EN | 0x40001090 | PA port interrupt enable register | 0x0000 |
| R16_PB_INT_EN | 0x40001092 | PB port interrupt enable register | 0x0000 |
| R16_PA_INT_MODE | 0x40001094 | PA port interrupt mode configuration register | 0x0000 |
| R16_PB_INT_MODE | 0x40001096 | PB port interrupt mode configuration register | 0x0000 |
| R16_PA_INT_IF | 0x4000109C | PA port interrupt flag register | 0x0000 |
| R16_PB_INT_IF | 0x4000109E | PB port interrupt flag register | 0x0000 |
| R32_PA_DIR | 0x400010A0 | PA port direction configuration register | 0x00000000 |
| R32_PA_PIN | 0x400010A4 | PA port pin input register | 0x0000XXXX |
| R32_PA_OUT | 0x400010A8 | PA port data output register | 0x00000000 |
| R32_PA_CLR | 0x400010AC | PA port data reset register | 0x00000000 |
| R32_PA_PU | 0x400010B0 | PA port pull-up resistor configuration register | 0x00000000 |
| R32_PA_PD_DRV | 0x400010B4 | PA port pull-down/drive configuration register | 0x00000000 |
| R32_PB_DIR | 0x400010C0 | PB port direction configuration register | 0x00000000 |
| R32_PB_PIN | 0x400010C4 | PB port pin input register | 0x00XXXXXX |
| R32_PB_OUT | 0x400010C8 | PB port data output register | 0x00000000 |
| R32_PB_CLR | 0x400010CC | PB port data reset register | 0x00000000 |
| R32_PB_PU | 0x400010D0 | PB port pull-up resistor configuration register | 0x00000000 |
| R32_PB_PD_DRV | 0x400010D4 | PB port pull-down/drive configuration register | 0x00000000 |

**Feature Pin Remapping Register (R16_PIN_ALTERNATE)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:12] | Reserved | RO | Reserved | 0000b |
| [11:9] | Reserved | RO | Reserved | 000b |
| 8 | RB_PIN_SPI0 | RW | SPI0 feature pin mapping select bit:<br>1: SCK0_/SCS_/MOSI_/MISO_ is mapped to PB[12]/PB[13]/PB[14]/PB[15];<br>0: SCK0/SCS/MOSI/MISO is mapped to PA[12]/PA[13]/PA[14]/PA[15]. | 0 |
| 7 | RB_PIN_UART3 | RW | UART3 feature pin mapping select bit:<br>1: RXD3_/TXD3_ is mapped to PB[20]/PB[21];<br>0: RXD3/TXD3 is mapped to PA[4]/PA[5]. | 0 |
| 6 | RB_PIN_UART2 | RW | UART2 feature pin mapping select bit:<br>1: RXD2_/TXD2_ is mapped to PB[22]/PB[23];<br>0: RXD2/TXD2 is mapped to PA[6]/PA[7]. | 0 |
| 5 | RB_PIN_UART1 | RW | UART1 feature pin mapping select bit:<br>1: RXD1_/TXD1_ is mapped to PB[8]/PB[9];<br>0: RXD1/TXD1 is mapped to PA[8]/PA[9]. | 0 |
| 4 | RB_PIN_UART0 | RW | UART0 feature pin mapping select bit:<br>1: RXD0_ /TXD0_ is mapped to PA[15]/PA[14];<br>0: RXD0/TXD0 is mapped to PB[4]/PB[7]. | 0 |
| 3 | RB_PIN_TMR3 | RW | TMR3 feature pin mapping select bit:<br>1: TMR3_ /PM3_ /CAP3_ is mapped to PB[18];<br>0: TMR3/PWM3/CAP3 is mapped to PA[2]. | 0 |
| 2 | RB_PIN_TMR2 | RW | TMR2 feature pin mapping select bit:<br>1: TMR2_/PM2_ /CAP2_ is mapped to PB[11];<br>0: TMR2/PWM2/CAP2 is mapped to PA[11]. | 0 |
| 1 | RB_PIN_TMR1 | RW | TMR1 feature pin mapping select bit:<br>1: TMR1_/PWM1_/CAP1_ is mapped to PB[10];<br>0: TMR1/PWM1/CAP1 is mapped to PA[10]. | 0 |
| 0 | RB_PIN_TMR0 | RW | TMR0 feature pin mapping select bit:<br>1: TMR0_/PWM0_/CAP0_ is mapped to PB[19];<br>0: TMR0/PWM0/CAP0 is mapped to PA[3]. | 0 |

**Peripheral Analog Pin Configuration Register (R16_PIN_ANALOG_IE)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 15 | RB_PIN_ADC4_5_IE | RW | ADC/TKEY 4/5 Channel Pin Digital Input Disabled:<br>1: Turn off the PA14-15 digital input to save power;<br>0: Turn on the digital input. | 0 |
| 14 | RB_PIN_ADC2_3_IE | RW | ADC/TKEY 2/3 Channel Pin Digital Input Disabled:<br>1: Turn off the PA12-13 digital input to save power;<br>0: Turn on the digital input. | 0 |
| 13 | RB_PIN_XT32K_IE | RW | 32KHz Crystal LSE Pin Digital Input Disabled:<br>1: Turn off the PA10-11 digital input to save power;<br>0: Turn on the digital input. | 0 |
| 12 | RB_PIN_ADC12_13_IE | RW | ADC/TKEY 12/13 Channel Pin Digital Input Disabled:<br>1: Turn off the PA8-9 digital input to save power; | 0 |

| | | | | |
|---|---|---|---|---|
| | | | 0: Turn on the digital input. | |
| 11 | RB_PIN_ADC10_11_IE | RW | ADC/TKEY 10/11 Channel Pin Digital Input Disabled:<br>1: Turn off the PA6-7 digital input to save power;<br>0: Turn on the digital input. | 0 |
| 10 | RB_PIN_ADC0_1_IE | RW | ADC/TKEY 0/1 Channel Pin Digital Input Disabled:<br>1: Turn off the PA4-5 digital input to save power;<br>0: Turn on the digital input. | 0 |
| 9 | RB_PIN_ADC6_7_IE | RW | ADC/TKEY 7/6 Channel Pin Digital Input Disabled:<br>1: Turn off the PA2-3 digital input to save power;<br>0: Turn on the digital input. | 0 |
| 8 | RB_PIN_ADC8_9_IE | RW | ADC/TKEY 9/8 Channel Pin Digital Input Disabled:<br>1: Turn off the PA0-1 digital input to save power;<br>0: Turn on the digital input. | 0 |
| 7 | RB_PIN_USB_IE | RW | USB Pin Enabled:<br>1: PB10-11 are USB communication pins;<br>0: PB10-11 are not used for USB communication. | 0 |
| 6 | RB_PIN_ETH_IE | RW | ETH Pin Enabled:<br>1: PB12-15 are the ETH communication pins. If LCD is not enabled entirely, RB_PIN_SEG12_15_IE can be set to 1 to turn off the digital input, which can save power.<br>0: PB12-15 are not used for ETH communication. | 0 |
| 5 | RB_PIN_SEG20_23_IE | RW | LCD 20-23 Segment Pin Enabled/Digital Input Disabled:<br>1: PB20-23 are segment drive. Turn off the digital input;<br>0: Turn on the digital input and turn off the LCD segment drive. | 0 |
| 4 | RB_PIN_SEG16_19_IE | RW | LCD 16-19 Segment Pin Enable/Digital Input Disabled:<br>1: PB16-19 are segment drive. Turn off the digital input;<br>0: Turn on the digital input and turn off the LCD segment drive. | 0 |
| 3 | RB_PIN_SEG12_15_IE | RW | LCD 12-15 Segment Pin Enable/Digital Input Disabled:<br>1: PB12-15 are segment drive. Turn off the digital input;<br>0: Turn on the digital input and turn off the LCD segment drive. | 0 |
| 2 | RB_PIN_SEG8_11_IE | RW | LCD 8-11 Segment Pin Enable/Digital Input Disabled:<br>1: PB8-11 are segment drive. Turn off digital input;<br>0: Turn on the digital input and turn off the LCD segment drive. | 0 |
| 1 | RB_PIN_SEG4_7_IE | RW | LCD 4-7 Segment Pin Enabled / Digital Input Disabled:<br>1: PB4-7 are segment drive. Turn off the digital input;<br>0: Turn on the digital input and turn off the LCD | 0 |

| | | | | |
|---|---|---|---|---|
| | | | segment drive. | |
| 0 | RB_PIN_SEG0_3_IE | RW | LCD 0-3 Segment Pin Enabled/Digital Input Disabled:<br>1: PB0-3 are segment drive. Turn off the digital input;<br>0: Turn on the digital input and turn off the LCD segment drive. | 0 |

*Note: If a pin is used for analog feature (ADC/TouchKey/ETH), it is recommended to turn off the digital input feature of this pin (that is, set the digital input as disabled), thereby reducing power consumption and facilitating to reduce interference.*

**PA Port Interrupt Enable Register (R16_PA_INT_EN)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:0] | R16_PA_INT_EN | RW | PA pin interrupt enable bit:<br>1: Enable the corresponding interrupt;<br>0: Disable the corresponding interrupt. | 0000h |

**PB Port Interrupt Enable Register (R16_PB_INT_EN)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:0] | R16_PB_INT_EN | RW | PA pin interrupt enable bit:<br>1: Enable the corresponding interrupt;<br>0: Disable the corresponding interrupt. | 0000h |

**PA Port Interrupt Mode Configuration Register (R16_PA_INT_MODE)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:0] | R16_PA_INT_MODE | RW | PA pin interrupt mode selection bit:<br>1: Edge trigger; 0: Level trigger. | 0000h |

**PB Port Interrupt Mode Configuration Register (R16_PB_INT_MODE)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:0] | R16_PB_INT_MODE | RW | PB pin interrupt mode selection bit:<br>1: Edge trigger; 0: Level trigger. | 0000h |

**PA Port Interrupt Flag Register (R16_PA_INT_IF)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:0] | R16_PA_INT_IF | RW1 | PA pin interrupt flag bit. Write 1 and reset to zero:<br>1: Interrupted; 0: No interrupt. | 0000h |

**PB Port Interrupt Flag Register (R16_PB_INT_IF)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:0] | R16_PB_INT_IF | RW1 | PB pin interrupt flag bit. Write 1 and reset to zero:<br>1: Interrupted; 0: No interrupt. | 0000h |

**PA Port Direction Configuration Register (R32_PA_DIR)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:16] | Reserved | RO | Reserved | 0000h |
| [15:8] | R8_PA_DIR_1 | RW | Configure the current input and output direction of PA pin:<br>1: The pin is in the output mode;<br>0: The pin is in the input mode. | 00h |
| [7:0] | R8_PA_DIR_0 | RW | | 00h |

**PA Port Pin Input Register (R32_PA_PIN)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:16] | Reserved | RO | Reserved | 0000h |
| [15:8] | R8_PA_PIN_1 | RO | The current level status of PA pin (this bit value is valid only when the corresponding bit of R32_PA_DIR is 0):<br>1: Pin input is at the high level;<br>0: Pin input is at the low level. | XXh |
| [7:0] | R8_PA_PIN_0 | RO | | XXh |

**PA Port Data Output Register (R32_PA_OUT)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:16] | Reserved | RO | Reserved | 0000h |
| [15:8] | R8_PA_OUT_1 | RW | When the corresponding bit of direction register R32_PA_DIR is 1:<br>Control the level status of PA pin output:<br>1: Output at high level;<br>0: Output at low level.<br>When the corresponding bit of direction register R32_PA_DIR is 0:<br>Control the interrupt polarity selection of PA pin:<br>1: High level/rising edge;<br>0: Low level/falling edge. | 00h |
| [7:0] | R8_PA_OUT_0 | RW | | 00h |

**PA Port Data Reset Register (R32_PA_CLR)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:16] | Reserved | RO | Reserved | 0000h |
| [15:8] | R8_PA_CLR_1 | WZ | PA data register reset control:<br>1: The corresponding bit data of R32_PA_OUT is reset to 0;<br>0: No effect. | 00h |
| [7:0] | R8_PA_CLR_0 | WZ | | 00h |

**PA Port Pull-up Resistor Configuration Register (R32_PA_PU)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:16] | Reserved | RO | Reserved | 0000h |
| [15:8] | R8_PA_PU_1 | RW | PA pin pull-up resistor enable control:<br>1: Turn on pull-up resistor;<br>0: Turn off pull-up resistor. | 00h |
| [7:0] | R8_PA_PU_0 | RW | | 00h |

**PA Port Pull-down/Drive Configuration Register (R32_PA_PD_DRV)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:16] | Reserved | RO | Reserved | 0000h |
| [15:8] | R8_PA_PD_DRV_1 | RW | When the corresponding bit of direction register R32_PA_DIR is 0:<br>PA pin pull-down resistor enable control:<br>1: Turn on pull-down resistor;<br>0: Turn off pull-down resistor.<br>When the corresponding bit of direction register R32_PA_DIR is 1:<br>Select PA pin current drive capability:<br>1: 20mA level;0: 5mA level. | 00h |
| [7:0] | R8_PA_PD_DRV_0 | RW | | 00h |

**PB Port Direction Configuration Register (R32_PB_DIR)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:24] | Reserved | RO | Reserved | 00h |

| | | | | |
|---|---|---|---|---|
| [23:16] | R8_PB_DIR_2 | RW | Configure the current input and output direction of PB pin:<br>1: The pin is in the output mode;<br>0: The pin is in the input mode. | 00h |
| [15:8] | R8_PB_DIR_1 | RW | | 00h |
| [7:0] | R8_PB_DIR_0 | RW | | 00h |

**PB Port Pin Input Register (R32_PB_PIN)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:24] | Reserved | RO | Reserved | 00h |
| [23:16] | R8_PB_PIN_2 | RO | The current level status of PB pin (this bit value is valid only when the corresponding bit of R32_PB_DIR is 0):<br>1: Pin input is at the high level;<br>0: Pin input is at the low level. | XXh |
| [15:8] | R8_PB_PIN_1 | RO | | XXh |
| [7:0] | R8_PB_PIN_0 | RO | | XXh |

**PB Port Data Output Register (R32_PB_OUT)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:24] | Reserved | RO | Reserved | 00h |
| [23:16] | R8_PB_OUT_2 | RW | When the corresponding bit of direction register R32_ PB_ DIR is 1:<br>Control the level status of PB pin output:<br>1: Output high level; 0: Output low level. | 00h |
| [15:8] | R8_PB_OUT_1 | RW | When the corresponding bit of direction register R32_ PB_ DIR is 1:<br>Control the level status of PB pin output:<br>1: Output high level; 0: Output low level.<br>When the corresponding bit of direction register R32_ PB_ DIR is 0:<br>Control the interrupt polarity selection of PB pin:<br>1: High level /rising edge;0: Low level/falling edge. | 00h |
| [7:0] | R8_PB_OUT_0 | RW | | 00h |

**PB Port Data Reset Register (R32_PB_CLR)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:24] | Reserved | RO | Reserved | 00h |
| [23:16] | R8_PB_CLR_2 | WZ | PB data register reset control:<br>1: The corresponding bit data of R32_PB_OUT is reset to 0;<br>0: No effect. | 00h |
| [15:8] | R8_PB_CLR_1 | WZ | | 00h |
| [7:0] | R8_PB_CLR_0 | WZ | | 00h |

**PB Port Pull-up Resistor Configuration Register (R32_PB_PU)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:24] | Reserved | RO | Reserved | 00h |
| [23:16] | R8_PB_PU_2 | RW | Control PB pin pull-up resistor enable:<br>1: Turn on pull-up resistor;<br>0: Turn off pull-up resistor. | 00h |
| [15:8] | R8_PB_PU_1 | RW | | 00h |
| [7:0] | R8_PB_PU_0 | RW | | 00h |

**PB Port Pull-down/Drive Configuration Register (R32_PB_PD_DRV)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:24] | Reserved | RO | Reserved | 00h |
| [23:16] | R8_PB_PD_DRV_2 | RW | When the corresponding bit of direction register R32_PB_DIR is 0:<br>Control PB pin pull-down resistor enable:<br>1: Turn on pull-down resistor;<br>0: Turn off pull-down resistor.<br>When the corresponding bit of direction register R32_PB_DIR is 1: | 00h |
| [15:8] | R8_PB_PD_DRV_1 | RW | | 00h |
| [7:0] | R8_PB_PD_DRV_0 | RW | | 00h |

| | | | | |
|---|---|---|---|---|
| | | | Select the drive capability of PB pin current:<br>1: 20mA level; 0: 5mA level. | |

## 7.5 GPIO pin mode configuration

Each GPIO can be configured into 5 modes, as indicated in the following table:

Table 7-9 Port configuration

| Mode | R32_Px_DIR | R32_Px_PU | R32_Px_PD_DRV |
|---|---|---|---|
| Floating input/ high-resistance input / analog input | 0 | 0 | 0 |
| Input with pull-up resistor | 0 | 1 | 0 |
| Input with pull-down resistor | 0 | 0 | 1 |
| Push-pull output, 5mA level drive capability | 1 | X | 0 |
| Push-pull output, 20mA level drive capability | 1 | X | 1 |

# Chapter 8 General Purpose Timer (TMRx)

## 8.1 TMRx introduction

A chip is provided with four 26-bit timers: TMR0, TMR1, TMR2, and TMR3, with a maximum timing of 2^26 clock cycles. It is applicable to a variety of circumstances, including measuring the pulse length of input signal (input capture) or generating an output waveform (PWM). In addition, TMR1 and TMR2 support the feature of DMA. Each timer is completely independent and can be synchronously operated together.

### 8.1.1 Main feature

- Four 26-bit timers: each timer has a maximum timing of 2^26 clock cycles.
- Support timer interrupts, in addition, TMR1 and TMR2 support DMA and interrupts.
- Support the feature of capture and measurement of length or cycle of input pulse.
- The feature of capture can be set to be capture of level change and hold time of high or low level.
- Support 26-bit PWM and adjust the duty ratio of PWM dynamically.

## 8.2 Register description

Table 8-1 List of TMR0-related registers

| Name | Access address | Description | Reset value |
|---|---|---|---|
| R8_ TMR0_ CTRL_ MOD | 0x40002000 | Mode setting register | 0x02 |
| R8_ TMR0_ INTER_ EN | 0x40002002 | Interrupt enable register | 0x00 |
| R8_ TMR0_ INT_ FLAG | 0x40002006 | Interrupt flag register | 0x00 |
| R8_ TMR0_ FIFO_ COUNT | 0x40002007 | FIFO count register | 0x0X |
| R32_ TMR0_ COUNT | 0x40002008 | Current count value register | 0x0XXXXXXX |
| R32_ TMR0_ CNT_ END | 0x4000200C | Count final value setting register | 0x0XXXXXXX |
| R32_ TMR0_ FIFO | 0x40002010 | FIFO register | 0x0XXXXXXX |

Table 8-2 List of TMR1-related registers

| Name | Access address | Description | Reset value |
|---|---|---|---|
| R8_ TMR1_ CTRL_ MOD | 0x40002400 | Mode setting register | 0x02 |
| R8_ TMR1_ CTRL_ DMA | 0x40002401 | DMA control register | 0x00 |
| R8_ TMR1_ INTER_ EN | 0x40002402 | Interrupt enable register | 0x00 |
| R8_ TMR1_ INT_ FLAG | 0x40002406 | Interrupt flag register | 0x00 |
| R8_ TMR1_ FIFO_ COUNT | 0x40002407 | FIFO count register | 0x0X |
| R32_ TMR1_ COUNT | 0x40002408 | Current count value register | 0x0XXXXXXX |
| R32_ TMR1_ CNT_ END | 0x4000240C | Count final value register | 0x0XXXXXXX |
| R32_ TMR1_ FIFO | 0x40002410 | FIFO register | 0x0XXXXXXX |
| R16_ TMR1_ DMA_ NOW | 0x40002414 | DMA current buffer zone address | 0x0000XXXX |
| R16_ TMR1_ DMA_ BEG | 0x40002418 | DMA start buffer zone address | 0x0000XXXX |
| R16_ TMR1_ DMA_ END | 0x4000241C | DMA end buffer zone address | 0x0000XXXX |

Table 8-3 List of TMR2-related registers

| Name | Access address | Description | Reset value |
|---|---|---|---|
| R8_ TMR2_ CTRL_ MOD | 0x40002800 | Mode setting register | 0x02 |
| R8_ TMR2_ CTRL_ DMA | 0x40002801 | DMA control register | 0x00 |

| | | | |
|---|---|---|---|
| R8_ TMR2_ INTER_ EN | 0x40002802 | Interrupt enable register | 0x00 |
| R8_ TMR2_ INT_ FLAG | 0x40002806 | Interrupt flag register | 0x00 |
| R8_ TMR2_ FIFO_ COUNT | 0x40002807 | FIFO count register | 0x0X |
| R32_ TMR2_ COUNT | 0x40002808 | Current count value register | 0x0XXXXXXX |
| R32_ TMR2_ CNT_ END | 0x4000280C | Count final value register | 0x0XXXXXXX |
| R32_ TMR2_ FIFO | 0x40002810 | FIFO register | 0x0XXXXXXX |
| R16_ TMR2_ DMA_ NOW | 0x40002814 | DMA current buffer zone address | 0x0000XXXX |
| R16_ TMR2_ DMA_ BEG | 0x40002818 | DMA start buffer zone address | 0x0000XXXX |
| R16_ TMR2_ DMA_ END | 0x4000281C | DMA end buffer zone address | 0x0000XXXX |

Table 8-4 List of TMR3-related registers

| Name | Access address | Description | Reset value |
|---|---|---|---|
| R8_ TMR3_ CTRL_ MOD | 0x40002C00 | Mode setting register | 0x02 |
| R8_ TMR3_ INTER_ EN | 0x40002C02 | Interrupt enable register | 0x00 |
| R8_ TMR3_ INT_ FLAG | 0x40002C06 | Interrupt flag register | 0x00 |
| R8_ TMR3_ FIFO_ COUNT | 0x40002C07 | FIFO count register | 0x0X |
| R32_ TMR3_ COUNT | 0x40002C08 | Current count value register | 0x0XXXXXXX |
| R32_ TMR3_ CNT_ END | 0x40002C0C | Count final value setting register | 0x0XXXXXXX |
| R32_ TMR3_ FIFO | 0x40002C10 | FIFO register | 0x0XXXXXXX |

**Mode Setting Register (R8_TMRx_CTRL_MOD) (x=0/1/2/3)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7:6] | RB_TMR_CAP_EDGE | RW | In capture mode, select the capture trigger mode:<br>00: No trigger;<br>01: Capture the time between any edge changes;<br>10: Capture the time between the falling edge and the falling edge;<br>11: Capture the time between the rising edge and the rising edge.<br>In count mode, select the count edge:<br>00: Not sample count;<br>01: Sample count to any edge;<br>10: Sample count to the falling edge;<br>11: Sample count to the rising edge. | 00b |
| [7:6] | RB_TMR_PWM_REPEAT | RW | In PWM mode, select the data duplication mode:<br>00: Repeat 1 time; 01: Repeat 4 times:<br>10: Repeat 8 times; 11: Repeat 16 times. | 00b |
| 5 | Reserved | RO | Reserved | 0 |
| 4 | RB_TMR_CAP_COUNT | RW | Sub-mode of RB_TMR_MODE_IN=1 input mode:<br>1: Count mode; 0: Capture mode. | 0 |
| 4 | RB_TMR_OUT_POLAR | RW | In PWM mode, output polarity setting bit:<br>1: Default at high level, active at low level;<br>0: Default at low level, active at high level. | 0 |
| 3 | RB_TMR_OUT_EN | RW | Timer output enable bit:<br>1: Output enabled; 0: Output disabled. | 0 |

| | | | | |
|---|---|---|---|---|
| 2 | RB_TMR_COUNT_EN | RW | Timer count enable bit:<br>1: Enable counting; 0: Disable counting. | 0 |
| 1 | RB_TMR_ALL_CLEAR | RW | Timer's FIFO/counter/interrupt flag cleared:<br>1: Force emptying and clearing;<br>0: No clearing. | 1 |
| 0 | RB_TMR_MODE_IN | RW | Timer mode setting bit:<br>1: Input mode (capture mode or counting mode);<br>0: Timing mode or PWM mode. | 0 |

**Interrupt Enable Register (R8_TMRx_INTER_EN) (x=0/1/2/3)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7:5] | Reserved | RO | Reserved | 000b |
| 4 | RB_TMR_IE_FIFO_OV | RW | FIFO Overflow (FIFO full in capture mode or FIFO empty in PWM mode) Interrupt Enable Bit:<br>1: Enable interrupt; 0: Disable interrupt. | 0 |
| 3 | RB_TMR_IE_DMA_END | RW | DMA End Interrupt Enable bit (TMR1/2 support only):<br>1: Enable interrupt; 0: Disable interrupt. | 0 |
| 2 | RB_TMR_IE_FIFO_HF | RW | FIFO Used More Than Half (FIFO>=4 in capture mode or FIFO<4 in PWM mode) Interrupt Enable Bit:<br>1: Enable interrupt; 0: Disable interrupt. | 0 |
| 1 | RB_TMR_IE_DATA_ACT | RW | Data Activation (it means that every time new data is captured in capture mode, and it means that value triggers the end of the active level in PWM mode) Interrupt Enable Bit:<br>1: Enable interrupt; 0: Disable interrupt. | 0 |
| 0 | RB_TMR_IE_CYC_END | RW | Cycle End (it refers to timeout in capture mode, and it refers to the end of the cycle in PWM mode and timing mode) Interrupt Enable Bit:<br>1: Enable interrupt; 0: Disable interrupt. | 0 |

**Interrupt Flag Register (R8_TMRx_INT_FLAG) (x=0/1/2/3)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7:5] | Reserved | RO | Reserved | 000b |
| 4 | RB_TMR_IF_FIFO_OV | RW1 | FIFO overflow (FIFO full in capture mode or FIFO empty in PWM mode) flag bit. Write 1 to reset:<br>1: Has overflown; 0: No overflow. | 0 |
| 3 | RB_TMR_IF_DMA_END | RW1 | DMA completion flag bit. Write 1 to reset:<br>1: Completed; 0: Not completed. | 0 |
| 2 | RB_TMR_IF_FIFO_HF | RW1 | FIFO used more than half (FIFO>=4 in capture mode or FIFO<4 in PWM mode) flag bit. Write 1 to reset:<br>1: FIFO has been used more than half:<br>0: The FIFO has not been used more than half. | 0 |
| 1 | RB_TMR_IF_DATA_ACT | RW1 | Data activation (it means that each time new data is captured in capture mode. And it means that value triggers the end of the active level in PWM mode) flag bit. Write 1 to reset:<br>1: Generated/used data; 0: Not generated/not used. | 0 |

| | | | | |
|---|---|---|---|---|
| 0 | RB_TMR_IF_CYC_END | RW1 | Cycle end (it refers to timeout in capture mode, and it refers to the end of the cycle in PWW mode and timing mode) flag bit. Write 1 to reset:<br>1: Timeout/cycle end; 0: No timeout/not ended. | 0 |

**FIFO Count Register (R8_TMRx_FIFO_COUNT) (x=0/1/2/3)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7:0] | R8_TMRx_FIFO_COUNT | RO | Data count in FIFO. The maximum value is 8. | 0x0X |

**Current Count Value Register (R32_TMRx_COUNT) (x=0/1/2/3)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:0] | R32_TMRx_COUNT | RO | The current count value of the counter. | 0XXXXXXXh |

**Final Count Value Setting Register (R32_TMRx_CNT_END) (x=0/1/2/3)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:0] | R32_TMRx_CNT_END | RW | In timer mode, the number of clocks in one timing cycle;<br>In PWM mode, the total number of clocks in one single cycle of PWM;<br>In capture mode, the number of capture timeout clocks.<br>Only the lower 26 bits are valid, with a maximum value of 67108863.<br>In count mode, final count value -2 (overflow).<br>*Note: Write operation on this register will automatically clear the value in register R32_TMRx_COUNT.* | 0XXXXXXXh |

**FIFO Register (R32_TMRx_FIFO) (x=0/1/2/3)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:0] | R32_TMRx_FIFO | RO/ WO | The FIFO data register. Only the lower 26 bits are valid. | 0XXXXXXXh |

**DMA Control Register (R8_TMRx_CTRL_DMA) (x=1/2) (only supported by TMR1/2)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7:3] | Reserved | RO | Reserved | 00000b |
| 2 | RB_TMR_DMA_LOOP | RW | DMA address cycle feature enable bit:<br>1: Enable address loop;<br>0: Disable address loop.<br>If the DMA address loop is enabled, when the DMA address is added to the end address of the setting, it will automatically circulate and point to the set first address. | 0 |
| 1 | Reserved | RO | Reserved | 0 |
| 0 | RB_TMR_DMA_ENABLE | RW | DMA feature enable bit:<br>1: Enable DMA; 0: Disable DMA. | 0 |

**DMA Current Buffer Zone Address (R16_TMRx_DMA_NOW) (x=1/2)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:0] | R16_TMRx_DMA_NOW | RO | The current address of the DMA data buffer zone.<br>It can be used to calculate the number of conversions.<br>The calculation method:<br>COUNT = (TMR_DMA_NOW-TMR_DMA_BEG) / 4. | XXXXh |

**DMA Start Buffer Zone Address (R16_TMRx_DMA_BEG) (x=1/2)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:0] | R16_TMRx_DMA_BEG | RW | The start address of the DMA data buffer zone. Only the lower 15 bits are valid. The address must be 4 bytes aligned. | XXXXh |

**DMA End Buffer Zone Address (R16_TMRx_DMA_END) (x=1/2)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:0] | R16_TMRx_DMA_END | RW | The end address of the DMA data buffer zone (not included). Only the lower 15 bits are valid. The address must be 4 bytes aligned. | XXXXh |

## 8.3 Function description and configuration

### 8.3.1 Timing and counting function

Each timer on chip supports a maximum timing of 2^26 clock cycles and performs an incremental count mode. If the system clock cycle is 32MHz, the maximum timing is: 31.25nS*2^26≈2S. Each timer has an independent interrupt.

The timing function is operated as follows:

(1) Set RB_TMR_ALL_CLEAR to clear R32_TMRx_COUNT and interrupt flags, etc.
(2) Set the Register R32_TMRx_CNT_END to the value that needs to be timed; The specific calculation method is: Time = Tsys * R32_TMRx_CNT_END;
(3) Reset RB_TMR_ALL_CLEAR, and reset the timing mode corresponding to RB_TMR_MODE_IN;
(4) Optional step. Set R8_TMRx_INTER_EN Register, and set RB_TMR_IE_CYC_END to open the timing cycle interrupt;
(5) Set RB_TMR_COUNT_EN of R8_TMRx_CTRL_MOD Register to start the counting of the timer.
(6) When the counting of R32_TMRx_COUNT is equal to R32_TMRx_CNT_END, the timing is completed. At this time, set RB_TMR_IF_CYC_END of R8_TMRx_INT_FLAG to 1, and write 1 to reset.

The counting function is operated as follows:

(1) Set the direction of the I/O pin corresponding to the count to input.
(2) Set the count overflow end value in R32_TMRx_CNT_END.
(3) Configure R8_TMRx_CTRL_MOD, set RB_TMR_MODE_IN and RB_TMR_CAP_COUNT corresponding to the count mode, clear the RB_TMR_ALL_CLEAR bit, select the sampling edge mode by RB_TMR_CAP_EDGE, set RB_ TMR_COUNT_EN to 1 to enable the counting function.
(4) Optional step, set the corresponding interrupt enable register bit if interrupts need to be enabled.
(5) R32_TMRx_COUNT stores the current count value, every time the count reaches the end value, RB_TMR_IE_CYC_END will be set to 1 and R32_TMRx_COUNT will be cleared to 0. If the interrupt is enabled, a hardware interrupt will be triggered.

### 8.3.2 PWM

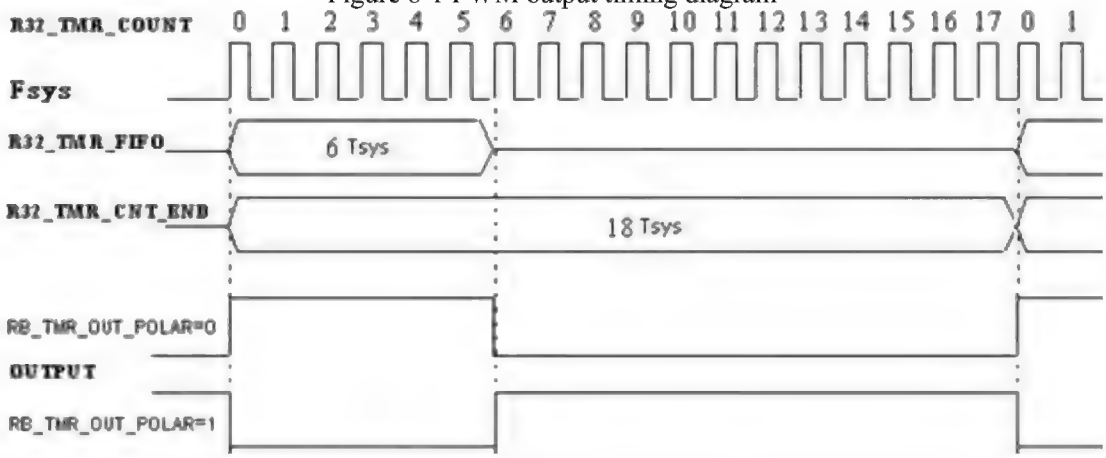
Each timer on chip has the feature of PWM. The PWM feature of TMR1 and TMR2 supports DMA data loading. The PWM can set the default output polarity to be high level or low level. The number of the same data duplication output can be selected as 1, 4, 8 or 16. This duplication feature combined with DMA can be used to simulate the effect of DAC. The shortest time unit of the PWM output active level is 1 system clock. The duty ratio of the PWM can be dynamically modified to simulate special waveforms.

PWM is operated at the following steps:

(1) Set RB_TMR_ALL_CLEAR, empty and reset R32_TMRx_FIFO, interrupt flags, etc.
(2) Set the PWM total cycle register R32_TMRx_CNT_END. This value should be no less than the value in the R32_TMRx_FIFO register;
(3) Configure R8_TMRx_CTRL_MOD, reset RB_TMR_ALL_CLEAR, reset the PWM mode corresponding to RB_TMR_MODE_IN, select the output polarity through RB_TMR_OUT_POLAR, and select the number of the same data duplications through RB_TMR_PWM_REPEAT as needed;
(4) Set data register R32_TMRx_FIFO. The minimum value is 0; the corresponding duty ratio is 0%; the maximum value is the same as R32_TMR_CNT_END; the corresponding duty ratio is 100%. The duty ratio is calculated as: R32_TMRx_FIFO/R32_TMRx_CNT_END. TMR1 and TMR2 can load continuous dynamic data through DMA and combine the number of the same data duplication output to simulate special waveforms;
(5) Configure R8_TMRx_CTRL_MOD, set RB_TMR_COUNT_EN to start counting and RB_TMR_OUT_EN to allow the output of PWM;
(6) Set the I/O pin corresponding to PWM as the output;
(7) Optional step. If an interrupt needs to be enabled, set the corresponding interrupt enable register bit;
(8) After one cycle of PWM is completed, if an interrupt is enabled, the hardware interrupt will be triggered after RB_TMR_IF_DATA_ACT or RB_TMR_IF_CYC_END is set.
(9) Update the data in R32_TMRx_FIFO to dynamically change the duty ratio of PWM. It is recommended to load by DMA.

For example, RB_TMR_OUT_POLAR bit is set to 0; R32_TMRx_FIFO is set to 6; R32_TMRx_CNT_END is set to 18. The basic timing diagram for generating PWM is shown below. The duty ratio is: R32_TMRx_FIFO / R32_TMRx_CNT_END = 1/3.

Figure 8-1 PWM output timing diagram



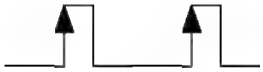

If RB_TMR_PWM_REPEAT is set to 00, it means that the above process is repeated once; 01 means repeating 4 times; 10 means repeating 8 times; 11 means repeating 16 times. After repetitions, continue to load the next data in the FIFO.

### 8.3.3 Capture

Each timer of the chip has the feature of capture. Specifically, the capture feature of TMR1 and TMR2 supports DMA data storage. Users can select from three capture modes: start from any edge trigger and end at any edge trigger, start from rising edge trigger and end at rising edge trigger, and start from falling edge trigger and end at falling edge trigger. The following table illustrates the modes of capture trigger:

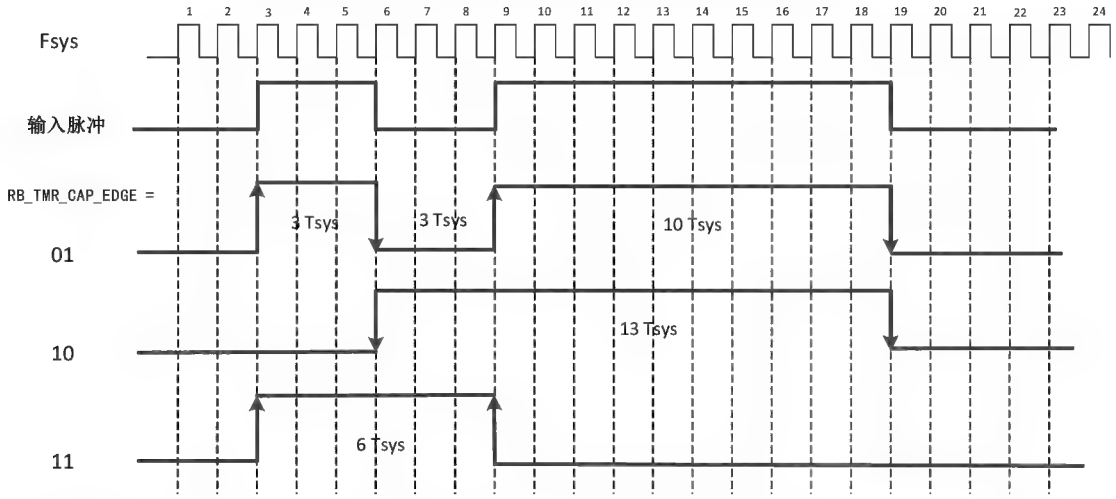
Table 8-5 Capture trigger modes

| Capture mode select bit RB_TMR_CATCH_EDGE | Trigger mode | Icon |
|---|---|---|
| 00 | No capture | None |
| 01 | Edge trigger<br>Edge to edge | |
| 10 | Falling edge to falling edge | |

| | | |
|---|---|---|
| 11 | Rising edge to rising edge |  |

Two trigger states in the edge trigger mode can capture the width at high level or the width at low level. If the highest bit (bit 25) of the valid data of the data register R32_TMRx_FIFO is 1, it indicates that the high level is captured, and 0 indicates that the low level is captured. If bit 25 of consecutive sets of data is 1 (or 0), it indicates that the width of the high (or low) level exceeds the timeout value and needs to be combined and accumulated.
In the trigger mode of falling edge to falling edge and rising edge to rising edge, an input change cycle can be captured. If the highest bit (bit 25) of the valid data of the data register R32_TMRx_FIFO is 0, it indicates that the normal sampling lasts one cycle, and 1 means that the input change cycle exceeds the timeout value of R32_TMRx_CNT_END. The latter group of data should be added and accumulated as a single input change cycle. The details are as follows:

Figure 8-2 Count the catch cycle with system clocks




As shown in the figure above, samples are taken once at each clock cycle:
When RB_TMR_CATCH_EDGE=01b, sampling is set to the mode of edge trigger. The width of sampled time is 3, 3, 10;
When RB_TMR_CATCH_EDGE=10b, sampling is set to the mode of falling edge to falling edge. The width of sampled time is 13;
When RB_TMR_CATCH_EDGE= 11b, sampling is set to the mode of rising edge to rising edge. The width of sampled time is 6.

Capture mode is operated in the following steps:
(1) Set RB_TMR_ALL_CLEAR, empty and reset R32_TMRx_FIFO, interrupt flags, etc.
(2) Set the I/O pin direction corresponding to capture as the input;
(3) Set a reasonable capture timeout time in R32_TMRx_CNT_END, which can be used to generate timeout interrupt when the input signal has not changed for a long time, or to generate timeout data after the input signal has no change in timeout (Bit 25 of data is 1. The lower 25 bits can be accumulated backwards);
(4) Configure R8_TMRx_CTRL_MOD, set the capture mode corresponding to RB_TMR_MODE_IN, select the captured edge mode by RB_TMR_CAP_EDGE, set RB_TMR_COUNT_EN of R8_TMRx_CTRL_MOD to 1 to enable counting;
(5) Optional step. If an interrupt needs to be enabled, set the corresponding interrupt enable register bit;
(6) If the captured data needs to be saved by means of DMA, set Register R16_TMRx_DMA_BEG as the first address of the buffer zone where the captured data is stored. Set Register R16_TMRx_DMA_END as the end address of the buffer zone where the captured data is stored (not included), and set RB_TMR_DMA_ENABLE of R8_TMRx_CTRL_DMA as 1 to enable the DMA feature;
(7) Reset the RB_TMR_ALL_CLEAR of R8_TMRx_CTRL_MOD to start the capture feature.
(8) Each time data is captured, RB_TMR_IF_DATA_ACT will be set to 1. If interrupt is enabled, the hardware interrupt will be triggered. The captured data is stored in R32_TMRx_FIFO by default. If DMA is enabled, the captured data will be automatically stored in the data buffer zone set in the DMA.

# Chapter 9 Universal Asynchronous Receiver Transmitter (UART)

## 9.1 UART introduction

A chip is provided with 4 sets of full-duplex asynchronous serial ports (UART0/1/2/3) and supports full-duplex and half-duplex serial communication. Specifically, UART0 provides transmit status pins for switching RS485 and supports MODEM signals including CTS, DSR, RI, DCD, DTR and RTS.

### 9.1.1 Main features

- Compatible with 16C550 asynchronous serial port and enhanced.
- Support 5, 6, 7 or 8 data bits and 1 or 2 stop bits.
- Support odd, even, no parity, blank 0, flag 1 and other verification methods.
- Programmable communication baud rate up to 5Mbps.
- Built-in 8-byte FIFO buffer supports 4 FIFO trigger stages.
- UART0 supports MODEM signals including CTS, DSR, RI, DCD, DTR, RTS.
- UART0 supports hardware flow control signals CTS, RTS automatic handshake and automatic transmission rate control, and is compatible with TL16C550C.
- Support the detection of serial frame errors and break circuit interval.
- Support full-duplex and half-duplex serial communication. UART0 provides transmit status pins for switching RS485.

## 9.2 Register description

Table 9-1 List of UART0-related registers

| Name | Access address | Description | Reset value |
|---|---|---|---|
| R8_UART0_MCR | 0x40003000 | MODEM control register | 0x00 |
| R8_UART0_IER | 0x40003001 | Interrupt enable register | 0x00 |
| R8_UART0_FCR | 0x40003002 | FIFO control register | 0x00 |
| R8_UART0_LCR | 0x40003003 | Line control register | 0x00 |
| R8_UART0_IIR | 0x40003004 | Interrupt identification register | 0x01 |
| R8_UART0_LSR | 0x40003005 | Line status register | 0x60 |
| R8_UART0_MSR | 0x40003006 | MODEM status register | 0xX0 |
| R8_UART0_RBR | 0x40003008 | Receive buffer register | 0xXX |
| R8_UART0_THR | 0x40003008 | Transmit hold register | 0xXX |
| R8_UART0_RFC | 0x4000300A | Receive FIFO count register | 0x00 |
| R8_UART0_TFC | 0x4000300B | Transmit FIFO count register | 0x00 |
| R16_UART0_DL | 0x4000300C | Baud rate divisor latch | 0xXX |
| R8_UART0_DIV | 0x4000300E | Prescaler divisor register | 0xXX |
| R8_UART0_ADR | 0x4000300F | Slave address register | 0xFF |

Table 9-2 List of UART1-related registers

| Name | Access address | Description | Reset value |
|---|---|---|---|
| R8_UART1_MCR | 0x40003400 | MODEM control register | 0x00 |
| R8_UART1_IER | 0x40003401 | Interrupt enable register | 0x00 |
| R8_UART1_FCR | 0x40003402 | FIFO control register | 0x00 |
| R8_UART1_LCR | 0x40003403 | Line control register | 0x00 |
| R8_UART1_IIR | 0x40003404 | Interrupt identification register | 0x01 |

| | | | |
|---|---|---|---|
| R8_UART1_LSR | 0x40003405 | Line status register | 0x60 |
| R8_UART1_RBR | 0x40003408 | Receive buffer register | 0xXX |
| R8_UART1_THR | 0x40003408 | Transmit hold register | 0xXX |
| R8_UART1_RFC | 0x4000340A | Receive FIFO count register | 0x00 |
| R8_UART1_TFC | 0x4000340B | Transmit FIFO count register | 0x00 |
| R16_UART1_DL | 0x4000340C | Baud rate divisor latch | 0xXX |
| R8_UART1_DIV | 0x4000340E | Prescaler divisor register | 0xXX |

Table 9-3 List of UART2-related registers

| Name | Access address | Description | Reset value |
|---|---|---|---|
| R8_UART2_MCR | 0x40003800 | MODEM control register | 0x00 |
| R8_UART2_IER | 0x40003801 | Interrupt enable register | 0x00 |
| R8_UART2_FCR | 0x40003802 | FIFO control register | 0x00 |
| R8_UART2_LCR | 0x40003803 | Line control register | 0x00 |
| R8_UART2_IIR | 0x40003804 | Interrupt identification register | 0x01 |
| R8_UART2_LSR | 0x40003805 | Line status register | 0x60 |
| R8_UART2_RBR | 0x40003808 | Receive buffer register | 0xXX |
| R8_UART2_THR | 0x40003808 | Transmit hold register | 0xXX |
| R8_UART2_RFC | 0x4000380A | Receive FIFO count register | 0x00 |
| R8_UART2_TFC | 0x4000380B | Transmit FIFO count register | 0x00 |
| R16_UART2_DL | 0x4000380C | Baud rate divisor latch | 0xXX |
| R8_UART2_DIV | 0x4000380E | Prescaler divisor register | 0xXX |

Table 9-4 List of UART3-related registers

| Name | Access address | Description | Reset value |
|---|---|---|---|
| R8_UART3_MCR | 0x40003C00 | MODEM control register | 0x00 |
| R8_UART3_IER | 0x40003C01 | Interrupt enable register | 0x00 |
| R8_UART3_FCR | 0x40003C02 | FIFO control register | 0x00 |
| R8_UART3_LCR | 0x40003C03 | Line control register | 0x00 |
| R8_UART3_IIR | 0x40003C04 | Interrupt identification register | 0x01 |
| R8_UART3_LSR | 0x40003C05 | Line status register | 0x60 |
| R8_UART3_RBR | 0x40003C08 | Receive buffer register | 0xXX |
| R8_UART3_THR | 0x40003C08 | Transmit hold register | 0xXX |
| R8_UART3_RFC | 0x40003C0A | Receive FIFO count register | 0x00 |
| R8_UART3_TFC | 0x40003C0B | Transmit FIFO count register | 0x00 |
| R16_UART3_DL | 0x40003C0C | Baud rate divisor latch | 0xXX |
| R8_UART3_DIV | 0x40003C0E | Prescaler divisor register | 0xXX |

**MODEM Control Register (R8_UARTx_MCR) (x=0/1/2/3)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| 7 | RB_MCR_HALF | RW | Half-duplex transceiver mode control (only supported by UART0):<br>1: Enter the half-duplex transceiver mode, transmit with priority, and receive when not transmitting;<br>0: Disable the half-duplex mode. | 0 |
| 6 | RB_MCR_TNOW | RW | Enable the status where DTR pin output is being transmitted (only supported by UART0):<br>1: Output the indication status of being transmitted to DTR pins so as to control the switching of RS485 transceiver;<br>0: DTR pins are in normal function. | 0 |
| 5 | RB_MCR_AU_FLOW_EN | RW | CTS and RTS hardware automatic flow control enable (only supported by UART0):<br>1: Enable CTS and RTS hardware automatic flow control;<br>0: Disable CTS and RTS hardware automatic flow control.<br>In the flow control mode, if this bit is 1, UART will continue to transmit the next data only when it detects that the CTS pin is valid (active at low level), otherwise it will suspend UART transmission. When this bit is 1, the change in CTS input status will not generate the MODEM status interrupt. If this bit is 1 and RTS is 1, when the received FIFO is empty, UART will automatically make the RTS pins valid (active at low level). Until the number of received bytes reaches the trigger point of the FIFO, UART will automatically invalidate the RTS pins and validate the RTS pins again when receiving empty FIFO. Hardware automatic flow control can be used to connect the CTS pins to the other's RTS pins and transmit the RTS pins to the other's CTS pins. | 0 |
| 4 | RB_MCR_LOOP | W | Enable the test mode of internal loop (only supported by UART0):<br>1: Enable the test mode of internal loop;<br>0: Disable the test mode of internal loop.<br>In the test mode of the internal loop, all external output pins of UART are inactive. TXD internally returns to RXD; RTS internally returns to CTS; DTR internally returns to DSR; OUT1 internally returns to RI; OUT2 internally returns to DCD. | 0 |
| 3 | RB_MCR_OUT2<br>RB_MCR_INT_OE | RW | UART interrupt request output control bit:<br>1: Allow sending request; 0: Disabled. | 0 |
| 2 | RB_MCR_OUT1 | RW | User-defined MODEM control bit (only supported by UART0). No actual output pin connected:<br>1: Set high; 0: Set low. | 0 |
| 1 | RB_MCR_RTS | RW | RTS signal output level control (only supported by UART0):<br>1: RTS signal output is active (at low level);<br>0: RTS signal output at high level (by default). | 0 |
| 0 | RB_MCR_DTR | RW | DTR signal output level control (only supported by UART0):<br>1: DTR signal output is active (at low level);<br>0: DTR signal output at high level (by default). | 0 |

**Interrupt Enable Register (R8_UARTx_IER) (x=0/1/2/3)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | RB_IER_RESET | WZ | UART Software Reset Control Bit. Automatically reset:<br>1: Software resets the serial port;<br>0: Normal operation. | 0 |

| | | | | |
|---|---|---|---|---|
| 6 | RB_IER_TXD_EN | RW | UART TXD Pin Output Enable Bit:<br>1: Enable pin output; 0: Disable pin output. | 0 |
| 5 | RB_IER_RTS_EN | RW | RTS Pin Output Enable Bit (only supported by UART0):<br>1: Enable pin output; 0: Disable pin output. | 0 |
| 4 | RB_IER_DTR_EN | RW | DTR Pin Output Enable Bit (only supported by UART0):<br>1: Enable pin output; 0: Disable pin output. | 0 |
| 3 | RB_IER_MODEM_CHG | RW | Modem Input Status Change Interrupt Enable Bit<br>(only supported by UART0):<br>1: Enable interrupt; 0: Disable interrupt. | 0 |
| 2 | RB_IER_LINE_STAT | RW | Transmit Line Status Interrupt Enable Bit:<br>1: Enable interrupt; 0: Disable interrupt. | 0 |
| 1 | RB_IER_THR_EMPTY | RW | Transmit Hold Register Empty Interrupt Enable Bit:<br>1: Enable interrupt; 0: Disable interrupt. | 0 |
| 0 | RB_IER_RECV_RDY | RW | Receive Data Interrupt Enable Bit:<br>1: Enable interrupt; 0: Disable interrupt. | 0 |

**FIFO Control Register (R8_UARTx_FCR) (x=0/1/2/3)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7:6] | RB_FCR_FIFO_TRIG | RW | Select the trigger points of receiving FIFO interrupt and hardware flow control:<br>00: 1 byte; 01: 2 bytes;<br>10: 4 bytes; 11: 7 bytes.<br>It is used to set the trigger points for receiving FIFO interrupt and hardware flow control. For example: 10 corresponds to 4 bytes. In other words, receiving 4 bytes can generate interrupts of receiving data and automatically invalidate RTS pins when the hardware flow control is enabled. | 00b |
| [5:3] | Reserved | RO | Reserved | 000b |
| 2 | RB_FCR_TX_FIFO_CLR | WZ | Transmit FIFO data clear enable bit. Automatically reset to zero:<br>1: Clear the data of the transmitted FIFO (excluding TSR);<br>0: Not clear the data of the transmitted FIFO. | 0 |
| 1 | RB_FCR_RX_FIFO_CLR | WZ | Receive FIFO data rest enable bit. Automatically reset to zero:<br>1: Clear the received FIFO data (RSR not included);<br>0: Not clear the received FIFO data. | 0 |
| 0 | RB_FCR_FIFO_EN | RW | FIFO enable bit:<br>1: Enable 8-byte FIFO; 0: Disable FIFO.<br>After FIFO is disabled, it is 16C450 compatible mode, that is, only one byte in FIFO (RECV_TG1=0, RECV_TG0=0, FIFO_EN=1). Enable is recommended. | 0 |

**Line Control Register (R8_UARTx_LCR) (x=0/1/2/3)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | RB_LCR_DLAB<br>RB_LCR_GP_BIT | RW | UART general-purpose bit, user-defined. | 0 |
| 6 | RB_LCR_BREAK_EN | RW | Force the BREAK line interval enable bit:<br>1: Forced generation; 0: Not generated. | 0 |

| | | | | |
|---|---|---|---|---|
| [5:4] | RB_LCR_PAR_MOD | RW | Select odd-even check bit format:<br>00: Odd parity check;<br>01: Even parity check;<br>10: Flag bit (MARK, set to 1);<br>11: Blank bit (SPACE, clear to 0).<br>Only valid when the RB_LCR_PAR_EN bit is 1. | 00b |
| 3 | RB_LCR_PAR_EN | RW | Odd-even check bit enable bit:<br>1: Allow odd-even check bits to be generated and received at the time of transmission;<br>0: No odd-even check bit. | 0 |
| 2 | RB_LCR_STOP_BIT | RW | Stop bit format setting bit:<br>0: One stop bit; 1: Two stop bits. | 0 |
| [1:0] | RB_LCR_WORD_SZ | RW | Select the length of serial data:<br>00: 5 data bits; 01: 6 data bits;<br>10: 7 data bits; 11: 8 data bits. | 00b |

**Interrupt Identification Register (R8_UARTx_IIR) (x=0/1/2/3)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7:6] | RB_IIR_FIFO_ID | RO | UART FIFO enable status bit:<br>11: FIFO has been enabled;<br>00: FIFO is not enabled. | 00b |
| [5:4] | Reserved | RO | Reserved | 00b |
| [3:0] | RB_IIR_INT_MASK | RO | Interrupt flag: If the RB_IIR_NO_INT bit is 0, it indicates that an interrupt is generated, and the interrupt source needs to be judged after reading. See Table 9-5 for details. | 0000b |
| 0 | RB_IIR_NO_INT | RO | UART no interrupt flag:<br>1: No interrupt; 0: Interrupt. | 1 |

The following table demonstrates the meanings of the RB_IIR_NO_INT bit of the Interrupt Identification Register R8_UARTx_IIR and each bit of RB_IIR_INT_MASK:

Table 9-5 Meaning of IR_IIR_INT_MASK in IIR Register

| IIR register bit | | | | Priority | Types of interrupt | Interrupt source | Method of clearing interrupts |
|---|---|---|---|---|---|---|---|
| IID3 | IID2 | IID1 | NOINT | | | | |
| 0 | 0 | 0 | 1 | None | No interrupt | No interruption | |
| 1 | 1 | 1 | 0 | 0 | Bus address matching | One data received is UART bus address, and the address matches the preset slave value or the broadcast address. (only supported by UART0) | Read IIR or disable multi-machine mode |
| 0 | 1 | 1 | 0 | 1 | Receiving line status | OVER_ERR, PAR_ERR, FRAM_ERR, BREAK_ERR | Read LSR |
| 0 | 1 | 0 | 0 | 2 | Receive data available | The number of bytes received reaches the trigger point of FIFO. | Read RBR |
| 1 | 1 | 0 | 0 | 2 | Receive data timeout | The next data is not received for more than 4 data hours. | Read RBR |
| 0 | 0 | 1 | 0 | 3 | THR register empty | Transmit hold register is empty, or the RB_IER_THR_EMPTY bit is changed from 0 to 1 and triggered. | Read IIR or write THR |
| 0 | 0 | 0 | 0 | 4 | MODEM input change | △CTS, △DSR, △RI, △DCD are set as 1 to trigger. | Read MSR |

**Line Status Register (R8_UARTx_LSR) (x=0/1/2/3)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | RB_LSR_ERR_RX_FIFO | O | Receive FIFO Error Flag Bit:<br>1: There is at least one PAR_ ERR, FRAM_ ERR or BREAK_ ERR error in the received FIFO;<br>0: There is no error in the received FIFO. | 0 |
| 6 | RB_LSR_TX_ALL_EMP | RO | Transmit Hold Register (THR) and Transmit Shift Register (TSR) Empty Flag Bit:<br>1: Both are empty;<br>0: Both are not empty. | 1 |
| 5 | RB_LSR_TX_FIFO_EMP | RO | Transmit FIFO Empty Flag Bit:<br>1: Transmit FIFO is empty;<br>0: Transmit FIFO is not empty. | 1 |
| 4 | RB_LSR_BREAK_ERR | RZ | BREAK Line Interval Detection Flag Bit:<br>1: BREAK is detected;<br>0: BREAK is not detected. | 0 |
| 3 | RB_LSR_FRAME_ERR | RZ | Data Frame Error Flag Bit:<br>1: There is a frame error in the data read from the FIFO being received, and a valid stop bit is missing;<br>0: There is no error in the currently read data frame. | 0 |
| 2 | RB_LSR_PAR_ERR | RZ | Receive Data Odd-even Check Error Flag Bit:<br>1: There is an odd-even check error in the data read from the FIFO being received;<br>0: The currently read data odd-even check is correct. | 0 |
| 1 | RB_LSR_OVER_ERR | RZ | Received FIFO Buffer Zone Overflow Flag Bit:<br>1: Has overflown:<br>0: No overflow. | 0 |
| 0 | RB_LSR_DATA_RDY | RO | There are Received Data in the Received FIFO Flag Bit:<br>1: There is data in the FIFO;<br>0: No data.<br>This bit is automatically reset to 0 after all data in the FIFO has been read. | 0 |

**MODEM status register (R8_UART0_MSR) (only supported by UART0)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | RB_MSR_DCD | RO | DCD pin status bit:<br>1: DCD pin is active (at low level);<br>0: DCD pin is inactive (at high level). | X |
| 6 | RB_MSR_RI | RO | RI pin status bit:<br>1: RI pin is active (at low level);<br>0: RI pin is inactive (at high level). | X |
| 5 | RB_MSR_DSR | RO | DSR pin status bit:<br>1: DSR pin is active (at low level);<br>0: DSR pin is inactive (at high level). | X |
| 4 | RB_MSR_CTS | RO | CTS pin status bit:<br>1: CTS pin is active (at low level);<br>0: CTS pin is invalid (at high level). | X |
| 3 | RB_MSR_DCD_CHG | RZ | DCD pin input status change flag bit:<br>1: Has changed; 0: No change. | 0 |
| 2 | RB_MSR_RI_CHG | RZ | RI pin input status change flag bit:<br>1: Has changed; 0: No change. | 0 |

| | | | | |
|---|---|---|---|---|
| 1 | RB_MSR_DSR_CHG | RZ | DSR pin input status change flag bit:<br>1: Has changed; 0: No change. | 0 |
| 0 | RB_MSR_CTS_CHG | RZ | CTS pin input status change flag bit:<br>1: Has changed; 0: No change. | 0 |

**Receive Buffer Register (R8_UARTx_RBR) (x=0/1/2/3)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7:0] | R8_UARTx_RBR | RO | Data reception buffer register.<br>If the DATA_RDY bit of LSR is 1, the received data can be read from this register;<br>If FIFO_EN is 1, the data received from the UART shift register (RSR) is first stored in the received FIFO and then read out via this register. | XXh |

**Transmit Hold Register (R8_UARTx_THR) (x=0/1/2/3)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7:0] | R8_UARTx_THR | RO | Transmit hold register.<br>It includes a transmit FIFO used for writing data to be transmitted. If FIFO_EN is 1, the written data is first stored in the transmitted FIFO and then output one by one via the transmit shift register (TSR). | XXh |

**Receive FIFO Count Register (R8_UARTx_RFC) (x=0/1/2/3)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7:0] | R8_UARTx_RFC | RO | The count of the data in the current received FIFO. | 00h |

**Transmit FIFO Count Register (R8_UARTx_TFC) (x=0/1/2/3)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7:0] | R8_UARTx_TFC | RO | The count of the data in the current transmitted FIFO. | 00h |

**Baud Rate Divisor Latch (R16_UARTx_DL) (x=0/1/2/3)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:0] | R16_UARTx_DL | RW | The 16-bit divisor is used to calculate the baud rate.<br>Formula: Divisor = UART Internal Reference Clock Fuart /16/ required communication baud rate.<br>Example: If the UART internal reference clock Fuart is 1.8432MHz and the required baud rate is 9600bps, Divisor = 1,843,200/16/9,600=12. | XXXXh |

**Prescaler Divisor Register (R8_UARTx_DIV) (x=0/1/2/3)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7:0] | R8_UARTx_DIV | RW | It is used to calculate the internal reference clock of UART. The low 7 bits are valid.<br>Formula: Divisor = Fsys*2 / Internal reference clock of UART. The maximum value is 127. | XXh |

**Slave Address Register (R8_UART0_ADR) (only supported by UART0)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7:0] | R8_UART0_ADR | RW | The slave address when UART0 is in multi-machine communication.<br>FFh: Not used; Other: Slave address. | 0FFh |

R8_UART0_ADR presets the local machine as the address of the slave, which is used to automatically compare with the received addresses during multi-machine communication and generate an interrupt when the addresses match or when the broadcast address 0FFH is received. The subsequent packets are allowed to be received. No data shall be received before the addresses match. Any data shall be stopped receiving after the data starts to be received or the R8_UART0_ADR Register is rewritten. Data is allowed to be received until the next address matches again or when the broadcast address is received.
When R8_UART0_ADR is 0FFH or RB_LCR_PAR_EN=0, the function of bus address auto compare is disabled. When R8_UART0_ADR is not 0FFH and RB_LCR_PAR_EN=1, the function of bus address auto compare is enabled. At the same time, the following parameters should be configured: RB_LCR_WORD_SZ is 11b, and the method of 8 data bits is selected. For the case where the address byte is MARK (i.e., Bit 9 of the data byte is 0), RB_LCR_PAR_MOD should be set to 10b. For the case where the address byte is SPACE (i.e., Bit 9 of the data byte is 1), RB_LCR_PAR_MOD should be set to 11b.

## 9.3 Function description and configuration

UART0/1/2/3 output pins are all 3.3V LVCMOS levels. The pins in UART mode include: data transmission pins and MODEM contact signal pins (only supported by UART0). Data transmission pins include TXD pins and RXD pins, and default at high level. MODEM contact signal pins include CTS pins, DSR pins, RI pins, DCD pins, DTR pins and RTS pins, and default at high level. All of these MODEM contact signals can be used as general purpose I/O pins, which are controlled by the applications and defined for their purpose.
Each of the four UARTs has an independent transceiver buffer zone and an 8-byte FIFO that supports simplex, half-duplex or full-duplex asynchronous serial communication. Serial data includes 1 low-level start bit, 5, 6, 7 or 8 data bits, 0 or 1 additional check bit or flag bit, 1 or 2 high level stop bit and supports odd check/even check/flag check/blank check. The baud rate error of UART transmission signals is less than 0.5%, and the allowable baud rate error of UART receiving signals is not more than 2%.

### 9.3.1 Baud rate calculation

1) Calculate UART internal reference clock Fuart, and set the R8_UARTx_DIV Register. The maximum value is 127, and normally 1 is written.
2) Calculate the baud rate and set the R16_UARTx_DL Register.

Baud Rate Formula = Fsys * 2 / R8_UARTx_DIV / 16 / R16_UARTx_DL.

### 9.3.2 UART transmission

"THR Register Empty" UART_II_THR_EMPTY interrupt sent by UART means that the current transmitted FIFO is empty. The interrupt is cleared when the IIR register is read or when the next data is written into the THR. If only one byte is written into THR, since the byte is quickly transferred to the transmit shift register (TSR) and starts to be transmitted, the request to send the THR register empty interrupt will be generated again soon. At this time, the next data to be sent can be written in. When the data in the TSR register is completely removed, the UART transmission is completed in a real sense. At this time, RB_LSR_TX_ALL_EMP bit of the LSR Register becomes 1 and active.
In the mode of interrupt trigger, after the interrupt of UART transmit hold register (THR) empty is received, if FIFO is enabled, up to 8 bytes can be written into the THR register and FIFO at a time, and then the controller will automatically send in order; if FIFO is disabled, only one byte can be written at one time; if no data needs to be sent, exit directly (the interrupt is automatically cleared when IIR is read before).
In the mode of query, whether the transmitted FIFO is empty can be judged according to the RB_LSR_TX_FIFO_EMP bit of the LSR Register. When this bit is 1, the data can be written to THR Register and FIFO. If FIFO is enabled, a maximum of 8 bytes can be written at one time.
R8_UARTx_TFC Register can also be read to know the number of remaining data to be sent in the current FIFO. If it is not equal to 8, the data to be sent can continue to be written in FIFO. This method can save filling time.

### 9.3.3 UART reception

UART Receive Data Available Interrupt UART_II_RECV_RDY means that the number of existing data bytes in the received FIFO has reached or exceeded the FIFO trigger point selected by the setting of RB_FCR_FIFO_TRIG in the FCR Register. This interrupt is cleared when reading data from RBR makes the number of FIFO words below the trigger point of FIFO.

UART Receive Data Timeout Interrupt UART_II_RECV_TOUT means that there is at least one byte of data in the received FIFO, and the waiting time is equivalent to the time of receiving 4 data starting from the last time when UART receives data and from the last time when data is taken away by the system. The interrupt can be cleared after a new data is received again or after the RBR Register is read once. When the received FIFO is completely empty, the RB_LSR_DATA_RDY bit in LSR Register is 0. When there is data in the received FIFO, the RB_LSR_DATA_RDY bit is 1 and active.

In the mode of interrupt trigger, after UART Receive Data Timeout Interrupt is received, read R8_UARTx_RFC Register to query the remaining data count in the current FIFO and read all the data directly, or continuously query RB_LSR_DATA_RDY of LSR Register. If this bit is valid, read the data until this bit is invalid. After UART Receive Data Available Interrupt is received, read the RB_FCR_FIFO_TRIG at one time from the RBR Register to get the number of bytes, or read all data in the current FIFO according to the RB_LSR_DATA_RDY bit and R8_UARTx_RFC Register.

In the mode of query, the RB_LSR_DATA_RDY bit of LSR Register can be used to determine whether the received FIFO is empty, or the R8_UARTx_RFC Register can be read to obtain the current data count of FIFO and all the data received by UART.

### 9.3.4 Hardware flow control

Hardware flow control includes automatic CTS (RB_MCR_AU_FLOW_EN set to 1) and automatic RTS (RB_MCR_AU_FLOW_EN and RB_MCR_RTS are both set to 1).

If automatic CTS is enabled, CTS pins must be valid before UART sends data. UART transmitter detects CTS pins before sending the next data. When the status of CTS pins is valid, the transmitter sends the next data. In order to stop the transmitter from transmitting the following data, the CTS pins must be invalidated before the middle time of the last stop bit currently being transmitted. The feature of automatic CTS reduces the interrupt requested to the system. When hardware flow control is enabled, the change in the level of CTS pins does not trigger a MODEM interrupt because the controller automatically controls the transmitter based on the status of CTS pins. If automatic RTS is enabled, the output of RTS pins is valid only when there is enough space in the FIFO to receive data, and the output of RTS pins is invalid when the received FIFO is full. The output of RTS pins is valid if the data in the received FIFO is completely removed or emptied. When the trigger point of the received FIFO is reached (the number of bytes in the received FIFO is not less than the number of bytes set by the RB_FCR_FIFO_TRIG of the FCR Register), the output of RTS pins is invalid, and the other transmitter is allowed to send additional data after the RTS pins are invalid. Once the data of the received FIFO is emptied, the RTS pins are automatically reasserted, causing the other transmitter to resume transmission. If both automatic CTS and automatic RTS are enabled (both RB_MCR_AU_FLOW_EN and RB_MCR_RTS in the MCR register are 1), when its own RTS pins are connected to the other's CTS pins, unless there is enough space in the received FIFO of the other party, the other party will not send any data. Therefore, by virtue of this hardware flow control, FIFO overflow and timeout errors during UART reception can be avoided.

# Chapter 10 Serial Peripheral Interface (SPI)

## 10.1 SPI introduction

SPI is a type of full-duplex serial interface with a master and a number of slaves connected to the bus. At the same time, only one pair of master and slave is in communication. Usually a SPI interface consists of 4 pins: SPI Chip Select Pin (SCS), SPI Clock Pin (SCK), SPI Serial Data Pin MISO (Master Input/Slave Output Pin), and SPI Serial Data Pin MOSI (Master Output/Slave Input Pin).

### 10.1.1 Main features

A chip is provided with two SPI interfaces with the following features:

- SPI0 supports both the master mode and the slave mode. SPI1 only supports the master mode.
- Compatible with the specifications of SPI.
- Support the modes of data transmission: mode 0 and mode 3.
- 8-bit data transmission mode. The sequence of data bits is optional: the low bits of a byte are in front or the high bits are in front.
- The clock frequency can be up to half of the system main frequency Fsys.
- 8-byte FIFO.
- SPI0 slave mode supports the first byte as the command mode or the data stream mode.
- SPI0 supports DMA. The efficiency of data transmission is higher.

## 10.2 Register description

Table 10-1 List of SPI0-related registers

| Name | Access address | Description | Reset value |
|---|---|---|---|
| R8_SPI0_CTRL_MOD | 0x40004000 | SPI0 mode control register | 0x02 |
| R8_SPI0_CTRL_CFG | 0x40004001 | SPI0 configuration register | 0x00 |
| R8_SPI0_INTER_EN | 0x40004002 | SPI0 interrupt enable register | 0x00 |
| R8_SPI0_CLOCK_DIV<br>R8_SPI0_SLAVE_PRE | 0x40004003 | SPI0 master mode clock divider register<br>SPI0 slave mode preset data register | 0x10 |
| R8_SPI0_ BUFFER | 0x40004004 | SPI0 data buffer zone | 0xXX |
| R8_SPI0_ RUN_ FLAG | 0x40004005 | SPI0 working status register | 0x00 |
| R8_SPI0_ INT_ FLAG | 0x40004006 | SPI0 interrupt flag register | 0x40 |
| R8_SPI0_FIFO_COUNT | 0x40004007 | SPI0 transceiver FIFO count register | 0x00 |
| R16_SPI0_TOTAL_CNT | 0x4000400C | SPI0 transceiver data total length register | 0x0000 |
| R8_SPI0_ FIFO | 0x40004010 | SPI0 data FIFO register | 0xXX |
| R8_SPI0_FIFO_COUNT1 | 0x40004013 | SPI0 transceiver FIFO count register | 0x00 |
| R16_SPI0_DMA_NOW | 0x40004014 | SPI0 DMA buffer zone current address | 0xXXXX |
| R16_SPI0_DMA_BEG | 0x40004018 | SPI0 DMA buffer zone start address | 0xXXXX |
| R16_SPI0_DMA_END | 0x4000401C | SPI0 DMA buffer zone end address | 0xXXXX |

Table 10-2 List of SPI1-related registers

| Name | Access address | Description | Reset value |
|---|---|---|---|
| R8_SPI1_CTRL_MOD | 0x40004400 | SPI1 mode control register | 0x02 |
| R8_SPI1_CTRL_CFG | 0x40004401 | SPI1 configuration register | 0x00 |
| R8_SPI1_INTER_EN | 0x40004402 | SPI1 interrupt enable register | 0x00 |
| R8_SPI1_CLOCK_DIV | 0x40004403 | SPI1 master mode clock divider register | 0x10 |
| R8_SPI1_BUFFER | 0x40004404 | SPI1 data buffer zone | 0xXX |

| | | | |
|---|---|---|---|
| R8_SPI1_RUN_FLAG | 0x40004405 | SPI1 working status register | 0x00 |
| R8_SPI1_INT_FLAG | 0x40004406 | SPI1 interrupt flag register | 0x40 |
| R8_SPI1_FIFO_COUNT | 0x40004407 | SPI1 transceiver FIFO count register | 0x00 |
| R16_SPI1_TOTAL_CNT | 0x4000440C | SPI1 transceiver data total length register | 0x00 |
| R8_SPI1_FIFO | 0x40004410 | SPI1 data FIFO register | 0xXX |
| R8_SPI1_FIFO_COUNT1 | 0x40004413 | SPI1 transceiver FIFO count register | 0x00 |

**SPI Mode Control Register (R8_SPIx_CTRL_MOD) (x=0/1)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | RB_SPI_MISO_OE | RW | MISO pin output enable bit (can be used for data line switching direction in the 2-wire mode):<br>1: MISO output enabled;<br>0: MISO output disabled. | 0 |
| 6 | RB_SPI_MOSI_OE | RW | MOSI pin output enable bit:<br>1: MOSI output enabled;<br>0: MOSI output disabled. | 0 |
| 5 | RB_SPI _SCK_OE | RW | SCK pin output enable bit:<br>1: SCK output enabled;<br>0: SCK output disabled. | 0 |
| 4 | RB_SPI_FIFO_DIR | RW | FIFO direction setting bit:<br>1: Input mode (indicating the reception of data);<br>0: Output mode (indicating the transmission of data). | 0 |
| 3 | RB_SPI_SLV_CMD_MOD | RW | Select the first byte mode (only supported by SPI0) in SPI0 slave mode:<br>1: First byte command mode;<br>0: Data stream mode.<br>In the first byte command mode, after the first byte data is received when the SPI chip selection becomes valid, it will be regarded as the command code, and RB_SPI_IF_FST_BYTE will be set to 1. | 0 |
| 3 | RB_SPI_MST_SCK_MOD | RW | Select the clock idle mode in the master mode:<br>1: Mode 3 (SCK is at high level when idle);<br>0: Mode 0 (SCK is at low level when idle). | 0 |
| 2 | RB_SPI_2WIRE_MOD | RW | Select 2-wire or 3-wire SPI mode in the slave mode (SPI0 only, SP11 does not require this control bit):<br>1: 2-wire mode / half-duplex (SCK/MISO);<br>0: 3-wire mode / full duplex (SCK/MOSI/MISO). | 0 |
| 1 | RB_SPI_ALL_CLEAR | RW | FIFO/Counter/Interrupt Flag of SPI is reset to zero:<br>1: Force emptying and clearing; 0: No clearing. | 1 |
| 0 | RB_SPI_MODE_SLAVE | RW | Select SPI0 master-slave mode (only supported by SPI0):<br>1: Slave mode; 0: Master mode. | 0 |

**SPI Configuration Register (R8_SPIx_CTRL_CFG) (x=0/1)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | Reserved | RO | Reserved | 0 |

| | | | | |
|---|---|---|---|---|
| 6 | RB_SPI_MST_DLY_EN | RW | Input delay enable in the master mode:<br>1: Enabled: used for high speed applications such as SPI clocks close to half of Fsys;<br>0: Disabled: regular applications. | 0 |
| 5 | RB_SPI_BIT_ORDER | RW | Select SPI data bit sequence:<br>1: Low bits in front;<br>0: High bits in front. | 0 |
| 4 | RB_SPI_AUTO_IF | RW | The function of automatically clearing the flag bit RB_SPI_IF_BYTE_END when accessing BUFFER/FIFO is enabled:<br>1: Enabled; 0: Disabled. | 0 |
| 3 | Reserved | RO | Reserved | 0 |
| 2 | RB_SPI_DMA_LOOP | RW | DMA address loop function enable bit (only supported by SPI0):<br>1: Enable address loop;<br>0: Disable address loop.<br>If the DMA address loop is enabled, when the DMA address is added to the end address of the setting, the auto loop points to the set first address. | 0 |
| 1 | Reserved | RO | Reserved | 0 |
| 0 | RB_SPI_DMA_ENABLE | RW | DMA function enable bit (only supported by SPI0):<br>1: DMA enabled; 0: DMA disabled. | 0 |

**SPI Interrupt Enable Register (R8_SPIx_INTER_EN) (x=0/1)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | RB_SPI_IE_FST_BYTE | RW | In the first byte command mode of the slave mode, the first byte reception interrupt enable bit (only supported by SPI0);<br>1: Enable the first received byte interrupt;<br>0: Disable the first received byte interrupt. | 0 |
| [6:5] | Reserved | RO | Reserved | 00b |
| 4 | RB_SPI_IE_FIFO_OV | RW | FIFO overflow (FIFO is full when receiving or FIFO is empty when transmitting) interrupt enable bit (only supported by SPI0):<br>1: Enable interrupt; 0: Disable interrupt. | 0 |
| 3 | RB_SPI_IE_DMA_END | W | DMA end interrupt enable bit (only supported by SPI0):<br>1: Enable interrupt; 0: Disable interrupt. | 0 |
| 2 | RB_SPI_IE_FIFO_HF | RW | More than half of FIFO used interrupt enable bit:<br>1: Enable interrupt; 0: Disable interrupt. | 0 |
| 1 | RB_SPI_IE_BYTE_END | RW | SPI single byte transmission completion interrupt enable bit:<br>1: Enable interrupt; 0: Disable interrupt. | 0 |
| 0 | RB_SPI_IE_CNT_END | RW | SPI all bytes transmission completion interrupt enable bit:<br>1: Enable interrupt; 0: Disable interrupt. | 0 |

**SPI Master Mode Clock Divider Register (R8_SPIx_CLOCK_DIV) (x=0/1)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| [7:0] | R8_SPI_CLOCK_DIV | RW | Division factor in the master mode. The minimum value is 2.<br>Fsck = Fsys / division factor. | 10h |

**SPI Slave Mode Preset Data Register (R8_SPI0_SLAVE_PRE) (only supported by SPI0)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7:0] | R8_SPI_SLAVE_PRE | RW | Preset the first returned data in the slave mode.<br>It is used to receive the returned data after the first byte data. | 10h |

**SPI Data Buffer Zone (R8_SPIx_BUFFER) (x=0/1)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7:0] | R8_SPIx_BUFFER | RW | SPI data transmit and reception buffer zone. | XXh |

**SPI Working Status Register (R8_SPIx_RUN_FLAG) (x=0/1)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | RB_SPI_SLV_SELECT | RO | Chip selection status bit in the slave mode (only supported by SPI0):<br>1: Being selected; 0: No chip selection. | 0 |
| 6 | RB_SPI_SLV_CS_LOAD | RO | First loading status bit after chip is selected in the salve mode (only supported by SPI0):<br>1: Loading R8_ SPI0_ SLAVE_ PRE:<br>0: Not loaded yet or have completed. | 0 |
| 5 | RB_SPI_FIFO_READY | RO | FIFO ready status bit:<br>1: FIFO is ready (R16_SPIx_TOTAL_CNT is not 0, and FIFO is not full when receiving or FIFO is not empty when transmitting);<br>0: FIFO is not ready. | 0 |
| 4 | RB_SPI_SLV_CMD_ACT | RO | Command reception completion status bit in the slave mode, that is, the first byte data has been exchanged (only supported by SPI0):<br>1: The first byte has just been exchanged;<br>0: The first byte has not been exchanged or it is not the first byte. | 0 |
| [3:0] | Reserved | RO | Reserved | 0000b |

**SPI Interrupt Flag Register (R8_SPIx_INT_FLAG) (x=0/1)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | RB_SPI_IF_FST_BYTE | RW1 | In slave mode, the first byte received flag (only supported by SPI0):<br>1: The first byte has been received;<br>0: Not received. | 0 |
| 6 | RB_SPI_FREE | RO | Current SPI idle status bit:<br>1: Current SPI is idle;<br>0: The current SPI is not idle. | 1 |
| 5 | Reserved | RO | Reserved | 0 |

| | | | | |
|---|---|---|---|---|
| 4 | RB_SPI_IF_FIFO_OV | RW1 | FIFO overflow (FIFO is full when receiving or FIFO is empty when transmitting) flag bit. Write 1 to reset:<br>1: FIFO has overflown:<br>0: FIFO has not overflowed. | 0 |
| 3 | RB_SPI_IF_DMA_END | RW1 | DMA completion flag bit (only supported by SPI0). Write 1 to reset:<br>1: Completed; 0: Not completed. | X |
| 2 | RB_SPI_IF_FIFO_HF | RW1 | More than half of FIFO used (FIFO>=4 when receiving or FIFO<4 when transmitting) flag bit. Write 1 to reset:<br>1: More than half of FIFO has been used;<br>0: More than half of FIFO has not been used; | 0 |
| 1 | RB_SPI_IF_BYTE_END | RW1 | SPI single byte transfer completion flag bit. Write 1 to reset:<br>1: Single byte transfer has been completed;<br>0: Transfer has not been completed. | 0 |
| 0 | RB_SPI_IF_CNT_END | RW1 | SPI full byte transfer completion flag bit. Write 1 to reset:<br>1: All bytes have been transferred;<br>0: Transfer has not been completed. | 1 |

**SPI Transceiver FIFO Count Register (R8_SPIx_FIFO_COUNT) (x=0/1)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7:0] | R8_SPIx_FIFO_COUNT | RW | The current count of bytes in the FIFO. | 00h |

**SPI Transceiver FIFO Count Register (R8_SPIx_FIFO_COUNT1) (x=0/1)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7:0] | R8_SPIx_FIFO_COUNT1 | RW | The current count of bytes in the FIFO.<br>Same as R8_SPIx_FIFO_COUNT. | 00h |

**SPI Transceiver Data Total Length Register (R16_SPIx_TOTAL_CNT) (x=0/1)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:0] | R16_SPIx_TOTAL_CNT | RW | The total number of bytes of SPI data transceiving in the master mode. The lower 12 bits are valid. Up to 4095 bytes can be sent and received at one time when using DMA. Slave mode is not supported. | 0000h |

**SPI Data FIFO Register (R8_SPIx_FIFO) (x=0/1)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7:0] | R8_SPIx_FIFO | RO/WO | Data FIFO register. | XXh |

Register R8_SPIx_BUFFER and Register R8_SPIx_FIFO are all SPI data-related registers. The main differences are:

The read of R8_SPIx_BURRER is the last data exchanged through SPI and does not affect FIFO and R8_SPIx_FIFO_COUNT. When R8_SPIx_BUFFER is written in the master mode, the byte is directly sent, and the operation of writing in the slave mode is undefined;

The read of R8_SPIx_FIFO is the earliest exchanged data taken from the FIFO, which will reduce the FIFO and R8_SPIx_FIFO_COUNT.

To write R8_SPIx_FIFO, data is temporarily stored in the FIFO. In the slave mode, the external SPI master decides

when to take it. In the master mode, data transmission is automatically started when R16_SPIx_TOTAL_CNT is not 0.

**SPI0 DMA Buffer Zone Current Address (R16_SPI0_DMA_NOW)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:0] | R16_SPI0_DMA_NOW | RW | The current address of the DMA data buffer zone.<br>It can be used to calculate the number of conversions.<br>The calculation method:<br>COUNT = SPI0_DMA_NOW - SPI0_DMA_BEG. | XXXXh |

**SPI0 DMA Buffer Zone Start Address (R16_SPI0_DMA_BEG)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:0] | R16_SPI0_DMA_BEG | RW | The start address of the DMA data buffer zone. Only the lower 15 bits are valid. | XXXXh |

**SPI0 DMA Buffer Zone End Address (R16_SPI0_DMA_END)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:0] | R16_SPI0_DMA_END | RW | The end address of the DMA data buffer zone (not included). Only the lower 15 bits are valid. | XXXXh |

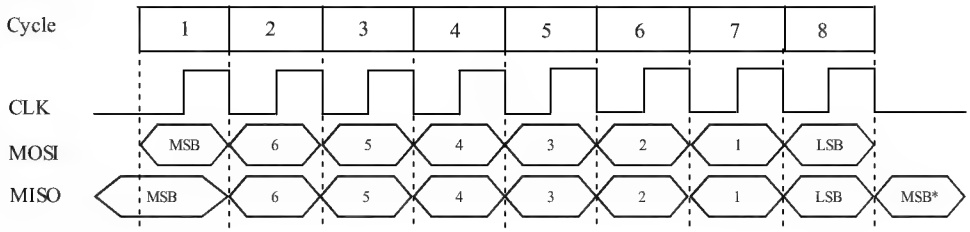
## 10.3 SPI transport format

SPI supports two transport formats: Mode 0 and Mode 3, which are selected by setting RB_SPI_MST_SCK_MOD of R8_SPIx_CTRL_MOD. Always sample and input serial data at SCK rising edge, and output serial data at falling edge.

The formats of data transmission are as shown below:

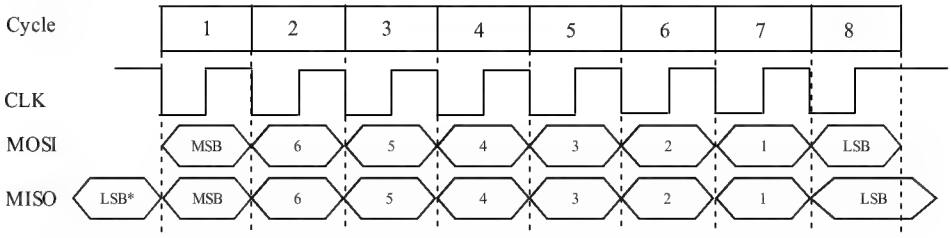
Mode 0: RB_SPI_MST_SCK_MOD = 0

Figure 10-1 Transport Format of SPI Mode 0




Mode 3: RB_SPI_MST_SCK_MOD = 1

Figure 10-2 Transport Format of SPI Mode 3



## 10.4 SPI configuration

### 10.4.1 SPI master mode

In the SPI master mode, a serial clock is generated on the SCK pins and the chip select pins can be designated as any I/O pin. Configuration steps are:

(1) Set R8_SPIx_CLOCK_DIV, and configure the frequency of SPI clocks;
(2) Set RB_SPI_MODE_SLAVE of R8_SPIx_CTRL_MOD to 0, and configure SPI in master mode;
(3) Set RB_SPI_MST_SCK_MOD of R8_SPIx_CTRL_MOD, and select clock idle mode 0 or mode 3;
(4) Set RB_SPI_FIFO_DIR of R8_SPIx_CTRL_MOD to configure the FIFO direction. If it is 1, FIFO is used for reception; if it is 0, FIFO is used for transmission.
(5) Set RB_SPI_MOSI_OE and RB_SPI_SCK_OE of R8_SPIx_CTRL_MOD to 1, RB_SPI_MISO_OE to 0, and set GPIO direction configuration register (R32_PA/PB_DIR) to make MOSI pins and SCK pins as the output and MISO pins as the input;
(6) In the 2-wire mode, SCK is unchanged. RB_SPI_MOSI_OE=0. MOSI is not used. Input (same as the 3-wire mode, RB_SPI_MISO_OE=0 and pins are set as the input) and output (RB_SPI_MISO_OE=1 and pins are set as the output) are implemented by MISO half-duplex. Directions are manually switched;
(7) Optional step. If DMA is enabled, R16_SPI_DMA_BEG should be written to the start address of the transceiver buffer zone, and R16_SPI_DMA_END should be written to the end address (not included). It is recommended to set RB_SPI_DMA_ENABLE after setting RB_SPI_FIFO_DIR. If it is confirmed that R16_SPIx_TOTAL_CNT is 0, RB_SPI_DMA_ENABLE can be first set to 1 to enable DMA feature.

Process of data transmission:

(1) Set RB_SPI_FIFO_DIR to 0 and the current FIFO direction as the output;
(2) Write the R16_SPIx_TOTAL_CNT Register, and set the length of the data to be sent;
(3) Write the R8_SPIx_FIFO Register and write the data to be sent to FIFO. If R8_SPIx_FIFO_COUNT is less than the FIFO capacity, continue to write FIFO. If DMA is enabled, DMA automatically loads the FIFO to complete this step.
(4) As long as R16_SPIx_TOTAL_CNT is not 0 and there is data in the FIFO, the SPI master will automatically send the data, otherwise it will pause;
(5) Wait for the R16_SPIx_TOTAL_CNT register to be 0, indicating that the data transmission is completed. If only one byte is sent, you can also query and wait for RB_SPI_FREE to be idle, or wait for R8_SPIx_FIFO_COUNT to be 0.

Process of data reception:

(1) Set RB_SPI_FIFO_DIR to 1 and the current FIFO direction as the input;
(2) Write the R16_SPIx_TOTAL_CNT register and set the length of the data to be received;
(3) As long as R16_SPIx_TOTAL_CNT is not 0 and the FIFO is not full, the SPI master will automatically receive the data, otherwise it will pause;
(4) Wait for the R8_SPIx_FIFO_COUNT register not to be 0, indicating that the return data is received. The value in the R8_SPIx_FIFO is the received data. If DMA is enabled, DMA automatically reads the FIFO to complete this step.

### 10.4.2 SPI slave mode

SPI0 supports the slave mode. In the slave mode, SCK pins are used to receive the serial clock of an externally connected SPI master. Configuration steps are:

(1) Set RB_SPI_MODE_SLAVE of R8_SPI0_CTRL_MOD to 1, and configure SPI0 to be slave mode;
(2) Set RB_SPI_SLV_CMD_MOD of R8_SPI0_CTRL_MOD as needed and select the slave first byte mode or the data stream mode;
(3) Set the RB_SPI_FIFO_DIR of R8_SPI0_CTRL_MOD to configure the FIFO direction (1: FIFO is used to receive; 0: FIFO is used to send);
(4) Set RB_SPI_MOSI_OE and RB_SPI_SCK_OE of R8_SPI0_CTRL_MOD to 0, RB_SPI_MISO_OE to 1, and set GPIO direction configuration register (R32_PA/PB_DIR) to make MOSI pins, SCK pins and SCS pins as the input, MISO pins as the input (the connection of multiple slaves is supported under the bus. After chip selection, MISO will automatically switch to output. One master and one slave are also supported) or the output (only for the connection of one master and one slave). In the SPI slave mode, the MISO's I/O pin direction also supports the automatic switching to output during the period when the SPI chip selection is valid, in addition to being able to be set as output by the GPIO direction configuration register. However, its output data

is selected by RB_SPI_MISO_OE (1: SPI data is the output; 0: The data of the GPIO data output register is the output). It is recommended to set the MISO pins as an input so that MISO does not output when chip selection is invalid, which is convenient for sharing the SPI bus during multi-machine operation;

(5) Optional step. Set the preset data register R8_SPI0_SLAVE_PRE in the SPI0 slave mode, which is used for the first automatic loading in the buffer zone for external output after chip selection. After 8 clocks have passed (that is, the first data byte is exchanged between the master and the slave), the controller obtains the first byte data (command code) sent by the external SPI master, and the external SPI master exchange obtains the preset data (status value) in R8_SPI0_SLAVE_PRE. Bit 7 of R8_SPI0_SLAVE_PRE will be automatically loaded onto the MISO pins during SCK low level after SPI chip select is valid. For SPI Mode 0 (CLK is at low level by default), if Bit 7 of R8_SPI0_SLAVE_PRE is preset, the external SPI master will be able to get the preset value of Bit 7 of R8_SPI0_SLAVE_PRE by querying the MISO pins when the SPI chip select is valid but has not yet transferred data. In this way, the value of Bit 7 of R8_SPI0_SLAVE_PRE can be quickly obtained by only validating the SPI chip select (usually, a busy status is provided to the master, so that the master can conduct quick query);

(6) Optional step. If DMA is enabled, R16_SPI_DMA_BEG should be written to the start address of the send/reception buffer zone, and R16_SPI_DMA_END should be written to the end address (not included). RB_SPI_DMA_ENABLE must be set after RB_SPI_FIFO_DIR is set.

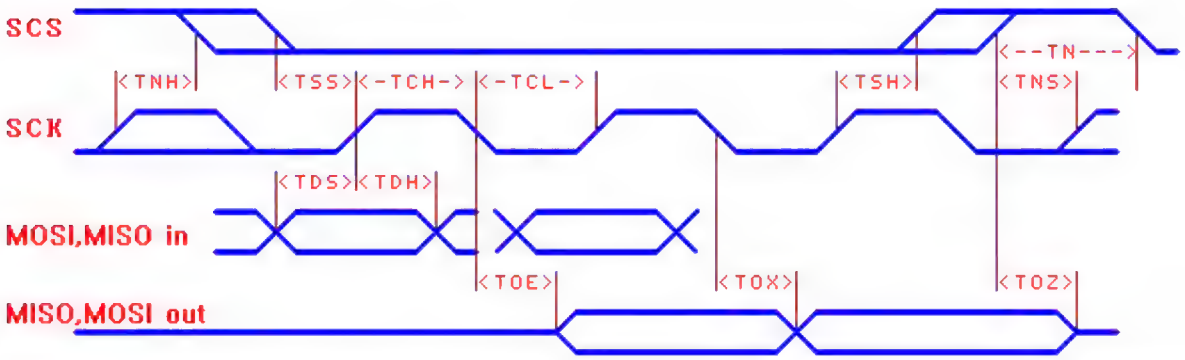
Process of data transmission:

(1) Set the RB_SPI_FIFO_DIR of R8_SPI0_CTRL_MOD to 0 and the current FIFO direction as the output;
(2) Optional step. If DMA is enabled, set RB_SPI_DMA_ENABLE to 1 and enable DMA function;
(3) Write multiple transmit data into the FIFO Register R8_SPI0_FIFO, and the time of taking away data is determined by the external SPI master. If DMA is enabled, DMA automatically loads FIFO to complete this step;
(4) Query R8_SPI0_FIFO_COUNT. If it is not full, continue to write the data to be sent to the FIFO.

Process of data acceptance:

(1) Set the RB_SPI_FIFO_DIR of R8_SPI0_CTRL_MOD to 1 and the current FIFO direction as the input;
(2) Optional step. If DMA is enabled, set RB_SPI_DMA_ENABLE to 1, and enable DMA function;
(3) Query R8_SPI0_FIFO_COUNT, if it is not empty, it indicates that the data has been received. The data is taken by reading R8_SPI0_FIFO. If DMA is enabled, the DMA automatically reads the FIFO to complete this step;
(4) For the reception of single byte data, R8_SPIx_BUFFER can also be directly read without using FIFO.

## 10.5 SPI timing




| Name | Parameter description (TA=25°C, VIO33=3.3V) | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|
| TSS | Setup time of valid SCS before SCK rising edge | Tsys*1.05 | | | nS |
| TSH | Hold time of valid SCS after SCK rising edge | Tsys*1.05 | | | nS |
| TNS | Setup time of invalid SCS before SCK rising edge | 15 | | | nS |
| TNH | Hold time of invalid SCS after SCK rising edge | 15 | | | nS |

| | | | | | |
|---|---|---|---|---|---|
| TN | Time of invalid SCS (interval time of SPI operation) | Tsys*2 | | | nS |
| TCH | Time of SCK clock at high level | Tsys*0.55 | | | nS |
| TCL | Time of SCK clock at low level | Tsys*0.55 | | | nS |
| TDS | Setup time of MOSI/MISO input before SCK rising edge | 8 | | | nS |
| TDH | Hold time of MOSI/MISO input after SCK rising edge | 5 | | | nS |
| TOE | SCK falling edge to MISO/MOSI output valid | 0 | | 18 | nS |
| TOX | SCK falling edge to MISO/MOSI output change | 0 | 5 | 16 | nS |
| TOZ | SCS invalid to MISO/MOSI output invalid | 2 | | 24 | nS |

*Note: Tsys is the clock cycle of system domain frequency (1/Fsys).*

# Chapter 11 PWM

## 11.1 PWM controller introduction

In addition to the 4-channel 26-bit PWM outputs provided by the timer, the system also provides 8-channel 8-bit PWM output (PWM4 to PWM11) with an adjustable duty ratio and 8 optional cycles as the fixed PWM period for easy operation.

## 11.2 Register Description

Table 11-1 List of PWMx-related registers

| Name | Access address | Description | Reset value |
|---|---|---|---|
| R8_ PWM_ OUT_ EN | 0x40005000 | PWMx output enable register | 0x00 |
| R8_ PWM_ POLAR | 0x40005001 | PWMx output polarity configuration register | 0x00 |
| R8_ PWM_ CONFIG | 0x40005002 | PWMx Configuration Control Register | 0x0X |
| R8_ PWM_ CLOCK_ DIV | 0x40005003 | PWMx clock divider register | 0x00 |
| R32_ PWM4_ 7_ DATA | 0x40005004 | PWM4/5/6/7 data holding register | 0xXXXXXXXX |
| R8_ PWM4_ DATA | 0x40005004 | PWM4 data holding register | 0xXX |
| R8_ PWM5_ DATA | 0x40005005 | PWM5 data holding register | 0xXX |
| R8_ PWM6_ DATA | 0x40005006 | PWM6 data holding register | 0xXX |
| R8_ PWM7_ DATA | 0x40005007 | PWM7 data holding register | 0xXX |
| R32_ PWM8_ 11_ DATA | 0x40005008 | PWM8/9/10/11 data holding register | 0xXXXXXXXX |
| R8_ PWM8_ DATA | 0x40005008 | PWM8 data holding register | 0xXX |
| R8_ PWM9_ DATA | 0x40005009 | PWM9 data holding register | 0xXX |
| R8_ PWM10_ DATA | 0x4000500A | PWM10 data holding register | 0xXX |
| R8_ PWM11_ DATA | 0x4000500B | PWM11 data holding register | 0xXX |

**PWMx Output Enable Register (R8_PWM_OUT_EN)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | RB_PWM11_OUT_EN | RW | PWM11 output enable bit:<br>1: Enable; 0: Disable. | 0 |
| 6 | RB_PWM10_OUT_EN | RW | PWM10 output enable bit:<br>1: Enable; 0: Disable. | 0 |
| 5 | RB_PWM9_OUT_EN | RW | PWM9 output enable bit:<br>1: Enable; 0: Disable. | 0 |
| 4 | RB_PWM8_OUT_EN | RW | PWM8 output enable bit:<br>1: Enable; 0: Disable. | 0 |
| 3 | RB_PWM7_OUT_EN | RW | PWM7 output enable bit:<br>1: Enable; 0: Disable. | 0 |
| 2 | RB_PWM6_OUT_EN | RW | PWM6 output enable bit:<br>1: Enable; 0: Disable. | 0 |
| 1 | RB_PWM5_OUT_EN | RW | PWM5 output enable bit:<br>1: Enable; 0: Disable. | 0 |

| | | | | |
|---|---|---|---|---|
| 0 | RB_PWM4_OUT_EN | RW | PWM4 output enable bit:<br>1: Enable; 0: Disable. | 0 |

**PWMx Output Polarity Configuration Register (R8_PWM_POLAR)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | RB_PWM11_POLAR | RW | PWM11 output polarity control bit:<br>1: Default at high level, valid at low level;<br>0: Default at low level, valid at high level; | 0 |
| 6 | RB_PWM10_POLAR | RW | PWM10 output polarity control bit:<br>1: Default at high level, valid at low level;<br>0: Default at low level, valid at high level; | 0 |
| 5 | RB_PWM9_POLAR | RW | PWM9 output polarity control bit:<br>1: Default at high level, valid at low level;<br>0: Default at low level, valid at high level; | 0 |
| 4 | RB_PWM8_POLAR | RW | PWM8 output polarity control bit:<br>1: Default at high level, valid at low level;<br>0: Default at low level, valid at high level; | 0 |
| 3 | RB_PWM7_POLAR | RW | PWM7 output polarity control bit:<br>1: Default at high level, valid at low level;<br>0: Default at low level, valid at high level; | 0 |
| 2 | RB_PWM6_POLAR | RW | PWM6 output polarity control bit:<br>1: Default at high level, valid at low level;<br>0: Default at low level, valid at high level; | 0 |
| 1 | RB_PWM5_POLAR | RW | PWM5 output polarity control bit:<br>1: Default at high level, valid at low level;<br>0: Default at low level, valid at high level; | 0 |
| 0 | RB_PWM4_POLAR | RW | PWM4 output polarity control bit:<br>1: Default at high level, valid at low level;<br>0: Default at low level, valid at high level; | 0 |

**PWMx Configuration Control Register (R8_PWM_CONFIG)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | RB_PWM10_11_STAG_EN | RW | PW10/11 interleaved output enable bit:<br>1: Interleaved output;<br>0: Independent output. | 0 |
| 6 | RB_PWM8_9_STAG_EN | RW | PW8/9 interleaved output enable bit:<br>1: Interleaved output;<br>0: Independent output. | 0 |
| 5 | RB_PWM6_7_STAG_EN | RW | PW6/7 interleaved output enable bit:<br>1: Interleaved output;<br>0: Independent output. | 0 |
| 4 | RB_PWM4_5_STAG_EN | RW | PW4/5 interleaved output enable bit:<br>1: Interleaved output;<br>0: Independent output. | 0 |
| [3:2] | RB_PWM_CYC_MOD | RW | Select the width of PWM data:<br>00: 8-bit data width; 01: 7-bit data width;<br>10: 6-bit data width; 11: 5-bit data width. | 00b |

| | | | | |
|---|---|---|---|---|
| 1 | RB_PWM_STAG_ST | RO | PWM interleave flag bit:<br>1: Indicating that PWM5/7/9/11 is allowed to output;<br>0: Indicating that PWM4/6/8/10 is allowed to output; | X |
| 0 | RB_PWM_CYCLE_SEL | RW | Select PWM cycle matching PWM data width:<br>1: 255/127/63/31 clock cycles;<br>0: 256/128/64/32 clock cycles. | 0 |

**PWMx Clock Divider Register (R8_PWM_CLOCK_DIV)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7:0] | R8_PWM_CLOCK_DIV | RW | PWM reference clock division factor.<br>Fpwm = Fsys / R8_PWM_CLOCK_DIV. | 00h |

**PWM Data Holding Register Set 1 (R32_PWM4_7_DATA)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:24] | R8_PWM7_DATA | RW | PWM7 data holding register. | XXb |
| [23:16] | R8_PWM6_DATA | RW | PWM6 data holding register. | XXb |
| [15:8] | R8_PWM5_DATA | RW | PWM5 data holding register. | XXb |
| [7:0] | R8_PWM4_DATA | RW | PWM4 data holding register. | XXb |

**PWM Data Holding Register Set 2 (R32_PWM8_11_DATA)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:24] | R8_PWM11_DATA | RW | PWM11 data holding register. | XXb |
| [23:16] | R8_PWM10_DATA | RW | PWM10 data holding register. | XXb |
| [15:8] | R8_PWM9_DATA | RW | PWM9 data holding register. | XXb |
| [7:0] | R8_PWM8_DATA | RW | PWM8 data holding register. | XXb |

## 11.3 PWM configuration

(1) Set Register R8_PWM_CLOCK_DIV and configure the reference clock frequency of PWM;
(2) Set the PWM output polarity configuration register R8_PWM_POLAR, and configure the output polarity corresponding to PWMx;
(3) Set the PWM configuration control register R8_PWM_CONFIG, and set the PWM mode, data bit width and period;
(4) Set the PWM output enable register R8_PWM_OUT_EN and enable the corresponding PWMx output;
(5) Calculate the data according to the required duty cycle, and write into the corresponding data holding register R8_PWMx_DATA;
(6) Set the PWM pin direction required in PWM4-PWM11 as the output. Optionally set the drive capability of the corresponding I/O.
(7) Update the data in R8_PWMx_DATA as needed and update the output duty cycle.

Adjust the duty cycle of the modifiable output PWM of R8_PWMx_DATA Register. The duty cycle is calculated

as:
Ncyc = RB_PWM_CYCLE_SEL ? (2^n-1) : (2^n)
(n = data bit width), Ncyc result is between 63 and 256.
Duty cycle of PWMx output active level = R8_PWMx_DATA / Ncyc * 100%
PWMx output frequency, Fpwmout = Fpwm / Ncyc = Fsys / R8_PWM_CLOCK_DIV / Ncyc
If a DC signal needs to be generated via PWM, the PWMx output end can be filtered by R/C and other circuits. It is recommended to use a two-stage RC with a time constant much greater than 4 / Fpwmout, or a first-stage RC with a time constant much greater than 100 / Fpwmout.

# Chapter 12 LED Screen Controller

## 12.1 LED controller introduction

A chip is provided with an LED screen control card interface and built-in 2-byte FIFO, supports DMA and interrupt, saves CPU processing time, and supports 1/2/4 data cable interface.

## 12.2 Register description

Table 12-1 List of LED-related registers

| Name | Access address | Description | Reset value |
|---|---|---|---|
| R8_LED_CTRL_MOD | 0x40006400 | LED Mode Configuration Register | 0x02 |
| R8_LED_CLOCK_DIV | 0x40006401 | LED Serial Clock Divider Register | 0x10 |
| R8_LED_STATUS | 0x40006404 | LED Status Register | 0xX0 |
| R16_LED_FIFO | 0x40006408 | LED Data FIFO Register | 0xXXXX |
| R16_LED_DMA_CNT | 0x40006410 | LED DMA Remaining Count Register | 0x0000 |
| R16_LED_DMA_MAIN | 0x40006414 | LED Main Buffer Zone DMA Address | 0xXXXX |
| R16_LED_DMA_AUX | 0x40006418 | LED Auxiliary Buffer Zone DMA Address | 0xXXXX |

**LED Mode Configuration Register (R8_LED_CTRL_MOD)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7:6] | RB_LED_CHAN_MOD | RW | Select LED channel mode:<br>00: LED0, single channel output;<br>01: LED0/1, dual channel output;<br>10: LED0~3, 4-channel output;<br>11: LED0~3, 4-channel output, where LED2/3 channel data is from auxiliary buffer zone. | 00b |
| 5 | RB_LED_IE_FIFO | RW | Enable half of FIFO count interrupt:<br>1: FIFO count <=2 interrupt trigger;<br>0: Interrupt is disabled. | 0 |
| 4 | RB_ LED_ DMA_ EN | RW | Enable LED DMA feature and DMA interrupt:<br>1: Enable; 0: Disable. | 0 |
| 3 | RB_ LED_ OUT_ EN | RW | Enable LED signal output:<br>1: Enable; 0: Disable. | 0 |
| 2 | RB_LED_OUT_POLAR | RW | LED data output polarity control bit:<br>1: Opposition: data 0 output 1 and data 1 output 0;<br>0: Straight-through: data 0 output 0 and data 1 output 1. | 0 |
| 1 | RB_LED_ALL_CLEAR | RW | Clear LED FIFO/counter/interrupt flag:<br>1: Force emptying and clearing:<br>0: No clearing. | 1 |
| 0 | RB_ LED_ BIT_ORDER | RW | Select LED serial data bit sequence:<br>1: Low bits are in front;<br>0: High bits are in front. | 0 |

**LED Serial Clock Divider Register (R8_LED_CLOCK_DIV)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| [7:0] | R8_LED_CLOCK_DIV | RW | LED output clock division factor.<br>LEDC frequency = Fsys / R8_LED_CLOCK_DIV. | 10h |

**LED Status Register (R8_LED_STATUS)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | RB_LED_IF_DMA_END | RW1 | DMA completion flag bit, write 1 to reset or write R16_LED_DMA_CNT to reset:<br>1: Has completed; 0: Not completed. | 0 |
| 6 | RB_LED_FIFO_EMPTY | RO | FIFO empty status bit:<br>1: FIFO is empty;<br>0: There is data in FIFO. | 1 |
| 5 | RB_LED_IF_FIFO | RW1 | Over half of FIFO counts interrupt flag bit, write 1 to reset or write R16_LED_FIFO to reset:<br>1: FIFO count <=2;<br>0: FIFO count >2. | 0 |
| 4 | RB_LED_CLOCK | 0 | Current LED clock signal level status:<br>1: High level; 0: Low level. | X |
| 3 | Reserved | RO | Reserved | 0 |
| [2:0] | RB_LED_FIFO_COUNT | RO | The value of byte count in the current FIFO must be an even number. | 000b |

**LED Data FIFO Register (R16_LED_FIFO)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:0] | R16_LED_FIFO | WO | LED data FIFO entry, 16-bit write. | XXXXb |

**LED DMA Remaining Count Register (R16_LED_DMA_CNT)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:0] | R16_LED_DMA_CNT | RW | The current DMA remaining words (16 bits) count of the LED_DMA_MAIN main buffer zone automatically decrements after DMA startup. Only the lower 12 bits are valid. The auxiliary buffer zone is not included. | 0000b |

**LED Main Buffer Zone DMA Address (R16_LED_DMA_MAIN)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:0] | R16_LED_DMA_MAIN | RW | DMA start address / current address of the main buffer zone automatically increment after the initial value is preset and they must be 2 bytes aligned. | XXXXb |

**LED Auxiliary Buffer Zone DMA Address (R16_LED_DMA_AUX)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:0] | R16_LED_DMA_AUX | RW | DMA start address / current address of the auxiliary buffer zone automatically increment after the initial value is preset and they must be 2 bytes aligned. | XXXXb |

## 12.3 LED configuration

(1) Set R8_LED_CLOCK_DIV to select the LED output clock frequency;
(2) Set R16_LED_DMA_MAIN to point to the buffer zone that prepares the output data, that is, the main buffer zone;
(3) If LED channel mode 3 is selected, set R16_LED_DMA_AUX to point to the auxiliary buffer zone;

(4) Set R8_LED_CTRL_MOD, select the channel mode, output polarity, bit order, enable interrupt and DMA feature, etc.;
(5) Set LEDC and necessary LED0~LED3 pin direction as the output, and optionally set the corresponding I /O drive capability;
(6) Set the DMA count register R16_LED_DMA_CNT, start DMA send, or send data by writing FIFO.

# Chapter 13 Segment LCD

## 13.1 Segment LCD control introduction

A chip is provided with a segment LCD controller interface that supports 96-point (24x4) LCD panels, adjustable drive voltage, adjustable scanning frequency, and LCD with specifications of 1/2 duty, 1/3 duty, 1/4 duty and 1/3 bias, and 1/2 bias.

## 13.2 Register description

Table 13-1 List of LCD-related registers

| Name | Access address | Description | Reset value |
|---|---|---|---|
| R8_LCD_CTRL_MOD | 0x40006000 | LCD Mode Configuration Register | 0x00 |
| R32_LCD_RAM0 | 0x40006004 | LCD Data Register Set 0 | 0x00000000 |
| R32_LCD_RAM1 | 0x40006008 | LCD Data Register Set 1 | 0x00000000 |
| R32_LCD_RAM2 | 0x4000600C | LCD Data Register Set 2 | 0x00000000 |

**LCD Mode Configuration Register (R8_LCD_CTRL_MOD)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | RB_LCD_V_SEL | WO | Select LCD drive voltage:<br>1: VIO33*76% (2.5V);<br>0: VIO33*100% (3.3V). | 0 |
| [6:5] | RB_LCD_SCAN_CLK | RW | Select LCD scan clock frequency:<br>00: 256Hz; 01: 512Hz;<br>10: 1KHz; 11: 128Hz. | 00b |
| [4:3] | RB_LCD_DUTY | RW | Select LCD driver duty:<br>00: 1/2 duty (COM0-COM1);<br>01: 1/3 duty (COM0-COM2);<br>10: 1/4 duty (COM0-COM3). | 00b |
| 2 | RB_LCD_BIAS | RW | LCD driver bias selection bit:<br>1: 1/3 bias; 0: 1/2 bias. | 0 |
| 1 | RB_LCD_POWER_ON | RW | LCD analog circuit enable bit:<br>1: Enable; 0: Disable. | 0 |
| 0 | RB_SYS_POWER_ON | RW | LCD logic circuit enable bit:<br>1: Enable; 0: Disable. | 0 |

**LCD Data Register Set 0 (R32_LCD_RAM0)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:0] | R32_LCD_RAM0 | RW | Data of segments SEG0-SEG7, 4 bits at each segment. | 00000000h |

**LCD Data Register Set 1 (R32_LCD_RAM1)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:0] | R32_LCD_RAM1 | RW | Data of segments SEG8-SEG15, 4 bits at each segment. | 00000000h |

**LCD Data Register Set 2 (R32_LCD_RAM2)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:0] | R32_LCD_RAM2 | RW | Data of segments SEG16-SEG23, 4 bits at each segment. | 00000000h |

## 13.3 Segment LCD configuration

(1) Select and turn on the 32KHz clock source;
(2) Set the LCD driver pins to be used as floating input. Optionally set R16_PIN_ANALOG_IE to save system power consumption. Note that external reset RST# multiplexes LCD driver pin SEG23. If SEG23 is to be used, cancel the feature of external reset;
(3) Load the segment code data to be displayed to the LCD data register R32_LCD_RAM0/1/2;
(4) Configure R8_LCD_CTRL_MOD, set the driving voltage, scanning frequency, bias, duty and other parameters, and set RB_LCD_POWER_ON and RB_SYS_POWER_ON, turn on the segment LCD driver;
(5) Update the LCD data register to display different contents at any time.

# Chapter 14 Passive Parallel Port

## 14.1 Passive parallel port introduction

A chip is provided with a passive parallel port with eight bidirectional data cables (PB7~PB0), four input control lines (SLVS/SLVW/SLVR/SLVA) and one output interrupt request line (SLVI). Among them, SLVS is chip select active at low level; SLVW/SLVR control write and read respectively, SLVA=0 means that data is operated; SLVA=1 means that a command or status is operated.

Table 14-1 Passive Parallel I/O Operation

| SLVS | SLVW | SLVR | SLVA | PB7~PB0 | Operation |
|---|---|---|---|---|---|
| 1 | X | X | X | Tri-state disable | Unselected, no action |
| 0 | 1 | 1 | X | Tri-state disable | Selected, but no action |
| 0 | 0 | 1/X | 1 | Input status | Write a command to CH579 with the command in R8_SLV_WR_DATA |
| 0 | 0 | 1/X | 0 | Input status | Write data to CH579 with the data in R8_SLV_WR_DATA |
| 0 | 1 | 0 | 1 | Output status | Read a status from CH579 with the status from R8_SLV_RD_STAT |
| 0 | 1 | 0 | 0 | Output status | Read data from CH579 with the data from R8_SLV_RD_DATA |

Register R8_SLV_WR_DATA is a multiplex of the R16_PB_INT_MODE high bytes. Register R8_SLV_RD_STAT and Register R8_SLV_RD_DATA are multiplexed with the R16_PB_INT_MODE low byte and R8_PB_OUT_0. Therefore, the feature of the multiplexed module will not be available after the passive parallel port is enabled.

The interrupt request output SLVI is the "logic OR" of R8_SLV_RD_STAT[7] and R8_PA_OUT_0[3]. When either of them is 1, output is at the high level by default. When both of them are 0, an interrupt request active at low level is output.

## 14.2 Register description

Table 14-2 List of registers related to passive parallel ports

| Name | Access address | Description | Reset value |
|---|---|---|---|
| R8_SLV_CONFIG | 0x4000101C | Passive parallel port configuration register | 0x00 |
| R8_SLV_RD_DATA | 0x400010C8 | Passive parallel port read data register | 0x00 |
| R8_SLV_RD_STAT | 0x40001096 | Passive parallel port read status register | 0x00 |
| R8_SLV_WR_DATA | 0x40001097 | Passive parallel port write data / command register | 0x00 |

**Passive parallel port configuration register (R8_SLV_CONFIG)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | RB_IF_SLV_RD | RW1 | Data read interrupt flag bit. Write 1 to reset to 0:<br>1: A read event is detected; 0: No event. | 0 |
| 6 | RB_IF_SLV_WR | RW1 | Data written interrupt flag bit. Write 1 to reset to 0:<br>1: A read event is detected; 0: No event. | 0 |
| 5 | RB_IF_SLV_CMD | RO | Command written flag:<br>1: A command is currently written;<br>0: No write-in currently or data is written. | 0 |
| 4 | Reserved | RO | Reserved | 0 |

| | | | | |
|---|---|---|---|---|
| 3 | RB_SLV_IE_RD | RW | Data read interrupt enable bit:<br>1: Enable; 0: Disable. | 0 |
| 2 | RB_SLV_IE_WR | RW | Data write interrupt enable bit:<br>1: Enable; 0: Disable. | 0 |
| 1 | RB_SLV_IE_CMD | RW | Command write interrupt enable bit:<br>1: Enable; 0: Disable. | 0 |
| 0 | RB_SLV_ENABLE | RW | Passive parallel port enable bit:<br>1: Enable; 0: Disable. | 0 |

## 14.3 Passive parallel port configuration

(1) Optional step. In order to enable the SLVI interrupt output, the status value must first be preset in R8_SLV_RD_STAT. Especially, R8_SLV_RD_STAT[7] associated with the SLVI interrupt pins should be set to 1. Clear R8_PA_OUT_0[ 3];
(2) Set the GPIO pins corresponding to SLVA (PA0), SLVS (PA1), SLVR (PB8) and SLVW (PB9) to the pull-up input (recommended) or the floating input mode, and set PB7~PB0 pins as the pull-up input (recommended) or the floating input mode;
(3) Set the RB_SLV_ENABLE of R8_SLV_CONFIG to enable the passive parallel port;
(4) Optional step. In order to enable the SLVI interrupt output, set the GPIO pins corresponding to SLVA (PA3) as the output;
(5) Optional step. Enable the passive parallel port interrupt response as required, combine RB_SLV_IE_RD/RB_SLV_IE_WR/RB_SLV_IE_CMD, and configure the corresponding channel enable of the core NVIC. It is recommended to enable only RB_SLV_IE_CMD;
(6) Optional step. The data to be sent is preset in R8_SLV_RD_DATA;
(7) Optional step. If the SLVI interrupt output is enabled and an interrupt request is required to output at low level via SLVI, the preset status value in R8_SLV_RD_STAT can be updated and R8_SLV_RD_STAT[7] is set to 0. If an interrupt request needs to be canceled, resume SLVI to the default high level, then set R8_SLV_RD_STAT[7] to 1 or set R8_PA_OUT_0[3] to 1;
(8) After the written command is interrupt or RB_IF_SLV_WR is queried as valid, when RB_IF_SLV_CMD is 1, the written command code is obtained from R8_SLV_WR_DATA. When RB_IF_SLV_CMD is 0, the written data is obtained from R8_SLV_WR_DATA;
(9) After RB_IF_SLV_RD is queried as valid, the next data or status can be preset.

# Chapter 15 Analog to Digital Converter (ADC)

## 15.1 ADC introduction

A chip is provided with a 12-bit successive approximation type analog-to-digital converter (ADC) that provides up to 16 channels and supports 14 external signal sources and 2 internal signal sources.

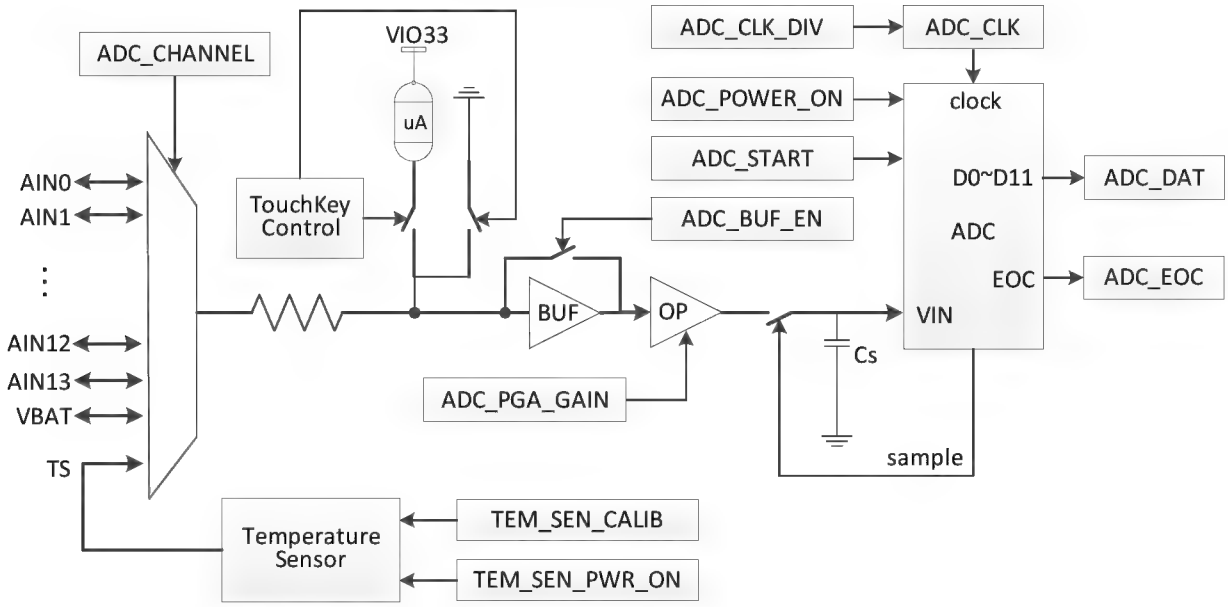
### 15.1.1 Main features

- 12-bit resolution.
- 14 external voltage sampling channels, internal temperature detection channels, internal battery voltage detection channel.
- Detection in the single-ended input mode and the differential input mode.
- The sampling clock frequency is optional.
- The range of ADC input voltage is 0V~VIO33.
- PGA is optional. Gain adjustment options are provided.
- Input buffer BUF is optional. High resistance signal sources are supported.

### 15.1.2 Function description

The figure below is a block diagram of an ADC module.

Figure 15-1 ADC structure diagram




## 15.2 Register description

Table 15-1 List of ADC-related registers

| Name | Access address | Description | Reset value |
|---|---|---|---|
| R8_ADC_CHANNEL | 0x40001058 | ADC input channel select register | 0x0F |
| R8_ADC_CFG | 0x40001059 | ADC configuration register | 0xA0 |
| R8_ADC_CONVERT | 0x4000105A | ADC conversion control register | 0x00 |
| R8_TEM_SENSOR | 0x4000105B | Temperature sensor control register | 0x04 |
| R16_ADC_DATA | 0x4000105C | ADC data register | 0x0XXX |
| R8_ADC_INT_FLAG | 0x4000105E | ADC interrupt flag register | 0x00 |

**ADC Input Channel Select Register (R8_ADC_CHANNEL)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7:4] | Reserved | RO | Reserved | 0000b |
| [3:0] | RB_ADC_CH_INX | RW | ADC channel index number. A total of 16 channels:<br>00h-0Dh: External signal channel AIN0~AIN13;<br>0Eh: Battery voltage VBAT;<br>0Fh: Built-in temperature sensor TS. | 1111b |

**ADC Configuration Register (R8_ADC_CFG)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7:6] | RB_ADC_CLK_DIV | RW | Select ADC clock frequency:<br>00: CK32M-based 10 fractional frequency, 3.2MHz;<br>01: CK32M-based 12 fractional frequency, 2.67MHz;<br>10: CK32M-based 6 fractional frequency, 5.33MHz;<br>11: CK32M-based 8 fractional frequency, 4MHz.<br>The actual sampling rate is approximately 1/16 of the clock frequency. | 10b |
| [5:4] | RB_ADC_PGA_GAIN | RW | Select ADC's input PGA gain:<br>00: -12dB, 1/4 times;<br>01: -6dB, 1/2 times;<br>10: 0dB, 1x, no gain;<br>11: 6dB, 2 times. | 10b |
| 3 | RB_ADC_OFS_TEST | RW | ADC offset error test mode:<br>1: Test mode (only supported by channel 1). In the test mode, the low 12 bits data of the data register R16_ADC_DATA will be inverted by bits (0x0579 is inverted as 0x0A86);<br>0: Normal mode. | 0 |
| 2 | RB_ADC_DIFF_EN | RW | ADC channel signal input mode:<br>1: Differential input; 0: Single-ended input. | 0 |
| 1 | RB_ADC_BUF_EN | RW | ADC Input buffer BUF enable:<br>1: On; 0: Off. | 0 |
| 0 | RB_ADC_POWER_ON | RW | ADC module power enable control:<br>1: Enable; 0: Disable. | 0 |

Table 15-2 PGA Gain Selection and Input Voltage Range Table

| PGA Gain Selection | The sampled voltage, Vi, is calculated from the ADC converted data | Upper limit of theoretically measurable voltage | Theoretically measurable voltage range (assuming Vref = 1.05V) | Recommended range of practical measurement voltage |
|---|---|---|---|---|
| -12dB (1/4 times) | (ADC/512-3)*Vref | 5*Vref | -0.2V ~ VIO33+0.2V | 2.9V ~ VIO33 |
| -6dB (1/2 times) | (ADC/1024-1)*Vref | 3*Vref | -0.2V ~ 3.15V | 1.9V ~ 3V |
| 0dB (1 times) | (ADC/2048)*Vref | 2*Vref | 0V ~ 2.1V | 0V ~ 2V |
| 6dB (2 times) | (ADC/4096+0.5)*Vref | 1.5*Vref | 0.525V ~ 1.575V | 0.6V ~ 1.5V |

ADC: The digital quantity after ADC conversion, namely R16_ADC_DATA.

Vref: The actual voltage value of the power supply node VINTA of the internal analog circuit is usually 1.05V±0.02V.

*Note: If sampling is taken at a low voltage after a negative gain (signal reduction), it may cause a large error in a voltage range; if sampling is taken at a high voltage after a positive gain (signal amplification), it may cause the conversion value of ADC to overflow. For this reason, it is recommended to select a reasonable gain mode based on the range of the measured signal voltage.*

*It is recommended to turn on the input buffer by default. The input buffer can be turned off to conduct ADC only when the internal resistance of the external signal source is less than 1KΩ.*

*When differential input is used, it is recommended to turn off the input buffer. When it is used for TouchKey detection, the input buffer must be turned on, and one of the two types of gains, either 0dB (priority) or -6dB, is recommended.*

**ADC Conversion Control Register (R8_ADC_CONVERT)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | RB_ADC_EOC_X | RO | ADC conversion end flag (asynchronous signal):<br>1: Complete; 0: In progress. | 0 |
| 6 | Reserved | RO | Reserved | 0 |
| 5 | RB_TKEY_CHG_ACT | RO | TouchKey capacitor charging status:<br>1: Capacitor charging; 0: Not charging. | 0 |
| 4 | RB_TKEY_ACTION | RO | TouchKey operation status:<br>1: Charging or discharging or ADC conversion;<br>0: Idle state. | 0 |
| 3 | RB_TKEY_PWR_ON | RW | TouchKey module power enable control:<br>1: Enable; 0: Disable. | 0 |
| [2:1] | Reserved | RO | Reserved | 00b |
| 0 | RB_ADC_START | RW | ADC conversion start control and status, automatic reset to zero:<br>1: Start conversion/converting;<br>0: Stop conversion. | 0 |

**Temperature Sensor Control Register (R8_TEM_SENSOR)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | RB_TEM_SEN_PWR_ON | RW | Control TS temperature sensor power enable:<br>1: Enable; 0: Disable. | 0 |
| [6:3] | Reserved | RW | Reserved | 0000b |
| [2:0] | RB_TEM_SEN_CALIB | RW | TS temperature sensor calibration value needs to remain unchanged. | 100b |

**ADC Data Register (R16_ADC_DATA)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:12] | Reserved | RW | Reserved | 0000b |
| [11:0] | RB_ADC_DATA | RO | Data after ADC conversion. | XXXh |

**ADC Interrupt Flag Register (R8_ADC_INT_FLAG)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | RB_ADC_IF_EOC | RO | ADC conversion completion interrupt flag.<br>This flag can be cleared by writing to register R8_ADC_CONVERT. | 0 |
| [6:0] | Reserved | RW | Reserved | 0000000b |

## 15.3 ADC configuration

### 15.3.1 External channel sampling

(1) Set RB_ADC_POWER_ON of R8_ADC_CFG to 1 to turn on the ADC, select the sampling frequency via RB_ADC_CLK_DIV, enable the input buffer and select the signal gains via RB_ADC_BUF_EN and RB_ADC_PGA_GAIN;
(2) Set R8_ADC_CHANNEL Register and select an external or internal signal channel;
(3) Set R8_ADC_CONVERT Register, set RB_ADC_START, and start ADC conversion;
(4) Query and wait for RB_ADC_START to automatically reset to zero or wait for RB_ADC_IF_EOC to be set to 1, indicating that the conversion is complete. Read R16_ADC_DATA to get 12-bit ADC conversion data. When there is enough time, it is recommended to convert and discard the first ADC data again;
(5) Repeat Step 2, Step 3 and Step 4. Continue to sample another channel or the next set of data.
(6) One ADC conversion cycle: ADC sampling (4 clocks) + conversion time (12 clocks) ≈ 16 Tadc. 1 time interval is added at time of continuous ADC, where Tadc = Tsys @ RB_ADC_CLK_DIV.
(7) If differential input is used:
Enable differential, select 0# channel: Actually conduct differential signal conversion on the voltage of AIN0 (positive end) and AIN2 (negative end);
Enable differential, select 1# channel: Actually conduct differential signal conversion on the voltage of AIN1 (positive end) and AIN3 (negative end);
In terms of the result after ADC conversion, if the data is greater than 0x800, it is indicated that the voltage of differential positive end is higher than that of differential negative end; if the data is less than 0x800, it is indicated that the voltage of differential positive end is lower than that of differential negative end. Taking PGA gain selection 0dB as an example, the theoretically measurable voltage range is -1.05V~1.05V. 0x400 means that the voltage of differential positive end is lower than that of differential negative end by about 0.5 Vref.

### 15.3.2 Temperature sensor sampling

(1) Set the RB_TEM_SEN_PWR_ON of the R8_TEM_SENSOR Register to 1 and turn on the temperature sensor, set R8_ADC_CHANNEL to 15, and select the temperature sensor signal to connect to the ADC input;
(2) Set RB_ADC_POWER_ON to 1 to turn on the ADC, reset RB_ADC_DIFF_EN, set RB_ADC_CLK_DIV, set RB_ADC_BUF_EN to 1, and set RB_ADC_PGA_GAIN to 0dB gain;
(3) Set the R8_ADC_CONVERT Register, set RB_ADC_START to 1, and start ADC conversion;
(4) Query and wait for RB_ADC_START to automatically reset to zero or wait for RB_ADC_IF_EOC to set as 1, read R16_ADC_DATA to obtain 12-bit ADC conversion data. In case of high requirements for accuracy, it is recommended to repeat Step 3 and Step 4 multiple times to calculate the average value of ADC data;
(5) The temperature value is obtained according to the conversion relationship of voltage and temperature. For details, please refer to the example program of evaluation board.

# Chapter 16 Touch-Key

## 16.1 Touch-Key introduction

A chip is provided with a capacitance detection module that can be used with the ADC module to implement the feature of capacitive touchkey detection. A total of 14 input channels support capacitance value of touchkey ranging from 10pF to 100pF.

## 16.2 Register description

Table 16-1 TouchKey Related Register List

| Name | Access address | Description | Reset value |
|---|---|---|---|
| R8_TKEY_CTRL | 0x4000105A | TouchKey Control Register | 0x00 |
| R8_TKEY_CNT | 0x4000105F | TouchKey Charge and Discharge Time Configuration Register | 0x00 |

Register R8_TKEY_CTRL is an alias for R8_ADC_CONVERT.

**TouchKey Control Register (R8_TKEY_CTRL)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | RB_ADC_EOC_X | RO | ADC conversion end flag:<br>1: Completed; 0: In progress. | 0 |
| 6 | Reserved | RO | Reserved | 0 |
| 5 | RB_TKEY_CHG_ACT | RO | TouchKey capacitor charging status:<br>1: Capacitor charging; 0: Not be charged. | 0 |
| 4 | RB_TKEY_ACTION | RO | TouchKey operation status:<br>1: Charging or discharging or ADC conversion; 0: Idle state. | 0 |
| 3 | RB_TKEY_PWR_ON | RW | TouchKey module power enable control: 1: Enable; 0: Disable. | 0 |
| [2:1] | Reserved | RO | Reserved | 00b |
| 0 | RB_ADC_START | RW | ADC conversion start control and status, automatic reset to zero:<br>1: Start conversion/converting;<br>0: Stop conversion. | 0 |

**TouchKey Charge and Discharge Time Configuration Register (R8_TKEY_CNT)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7:0] | R8_TKEY_CNT | WO | The number of charge and discharge cycles of Touch-Key (in Tsys).<br>The full 8 bits are the number of charge cycles.<br>Recommended not less than 0x10.<br>The number of discharge cycles is automatically generated according to the number of charge cycles.<br>When the number of charging cycles is less than 128, the number of discharging cycles is 8; when the number of charging cycles is greater than or equal to 128, the number of discharging cycles is 14. | 00h |

## 16.3 Touch-Key configuration

(1) Set RB_ADC_POWER_ON to 1 to turn on the ADC, reset RB_ADC_DIFF_EN, set RB_ADC_CLK_DIV, set RB_ADC_BUF_EN to 1, set RB_ADC_PGA_GAIN to 0dB or -6dB gain (the latter is used for smaller capacitance);
(2) Set RB_TKEY_PWR_ON of R8_TKEY_CTRL to 1, and enable the TouchKey module;
(3) Set R8_ADC_CHANNEL and select the channel where the target key is located;
(4) Optional step. If the interrupt mode is used, R8_TKEY_CTRL must be written to reset RB_ADC_IF_EOC the same as Step 2;
(5) Estimate the target key's capacitance, calculate its charging time, and write the number of charging cycles into the R8_TKEY_CNT register to calculate:
Number of charging cycles of R8_TKEY_CNT = ( Ckey + 5pF parasitic) * Vkey / Ikey / Tsys,
(Specifically, in case of 0dB gain, Vkey is recommended to be 1.6V; in case of -6dB gain, Vkey is recommended to be 2.4V; Ikey is about 35uA)
The external capacitance of the touch key is assumed as Ckey=35pF, Fsys=16MHz, so R8_TKEY_CNT=29. When the calculation result is less than 16, 16 is taken; when it is between 110 and 128, 128 is taken;
(6) Query and wait for RB_TKEY_ACTION to be 0 (or, if it is cleared before, it can wait for RB_ADC_IF_EOC to be set to 1), read R16_ADC_DATA to obtain 12-bit ADC conversion data, compare with the stored data of the same channel, analyze and judge whether the key is pressed. It is recommended to use software analysis to remove interference;
(7) Repeat Step 3, Step 4, Step 5 and Step 6. Continue detecting the keys of the next channel.
(8) If the interrupt mode is used, the following four steps will be repeated in the interrupt service program: 6: read data and make judgment; 4 or 2: clear interrupt flag; 3: select channel; 5: start detection.

# Chapter 17 USB Controller

## 17.1 USB controller introduction

A chip is embedded with the USB master-slave controller and transceiver. The features include:

- Support USB Host function and USB Device function.
- Support USB2.0 full-speed 12Mbps or low-speed 1.5Mbps.
- Support USB control transfer, batch transfer, interrupt transfer, sync/real-time transfer.
- Support up to 64-bytes of data packets, built-in FIFO, interrupts and DMA.

## 17.2 Register description

USB-related registers are divided into three categories, some of which are multiplexed in the host and device mode.

(1) USB global registers
(2) USB device control registers
(3) USB host control registers

### 17.2.1 Global register description

Table 17-1 List of USB-related registers (those marked as gray are reset and controlled by RB_UC_RESET_SIE)

| Name | Access address | Description | Reset value |
|---|---|---|---|
| R8_ USB_ TYPE_ C_ CTRL | 0x40008038 | USB Type-C Configuration Register | 0x00 |
| R8_ USB_ CTRL | 0x40008000 | USB Control Register | 0x06 |
| R8_ USB_ INT_ EN | 0x40008002 | USB Interrupt Enable Register | 0x00 |
| R8_ USB_ DEV_ AD | 0x40008003 | USB Device Address Register | 0x00 |
| R32_ USB_ STATUS | 0x40008004 | USB Status Register | 0xXX20XXXX |
| R8_ USB_ MIS_ ST | 0x40008005 | USB Miscellaneous Status Register | 0xXX |
| R8_ USB_ INT_ FG | 0x40008006 | USB Interrupt Flag Register | 0x20 |
| R8_ USB_ INT_ ST | 0x40008007 | USB Interrupt Status Register | 0xXX |
| R8_ USB_ RX_ LEN | 0x40008008 | USB Receive Length Register | 0xXX |

**USB Type-C Configuration Register**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | RB_UTCC_GP_BIT | RW | USB universal flag bit, user-defined. | 0 |
| 6 | RB_UCC2_PD_EN | RW | Internal 5.1K pull-down resistor for UCC2 pins:<br>1: Enable; 0: Disable. | 0 |
| [5:4] | RB_UCC2_PU_EN | RW | Internal pull-up resistor control bit for UCC2 pins:<br>00: Internal pull-up resistor is prohibited;<br>01: Turn on the internal 36K pull-up to provide the default USB current;<br>10: Turn on the internal 12K pull-up to provide 1.5A current;<br>11: Turn on the internal 4.7K pull-up to provide 3A current. | 00b |
| 3 | RB_VBUS_PD_EN | RW | Internal 10K pull-down resistor for VBUS pins:<br>1: Enable; 0: Disable. | 0 |
| 2 | RB_UCC1_PD_EN | RW | Internal 5.1K pull-down resistor for UCC1 pins:<br>1: Enable; 0: Disable. | 0 |

| | | | | |
|---|---|---|---|---|
| [1:0] | RB_UCC1_PU_EN | RW | Internal pull-up resistor control bit for UCC1 pins:<br>00: Internal pull-up resistor is prohibited;<br>01: Turn on the internal 36K pull-up to provide the default USB current;<br>10: Turn on the internal 12K pull-up to provide 1.5A current;<br>11: Turn on the internal 4.7K pull-up to provide 3A current. | 00b |

The above USB type-C pull-up resistor and pull-down resistor are independent of the port pull-up resistor controlled by the GPIO port direction control and pull-up resistor enable register. When a pin is used for USB type-C, the port pull-up resistor corresponding to this pin should be disabled. It is recommended to enable the high-impedance input mode for this pin (to avoid this pin output at low or high level).

**USB Control Register (R8_USB_CTRL)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | RB_UC_HOST_MODE | RW | USB working mode selection bit:<br>1: Host mode; 0: Device mode. | 0 |
| 6 | RB_UC_LOW_SPEED | RW | USB bus signal transmission rate selection bit:<br>1: 1.5Mbps: 0: 12Mbps. | 0 |
| 5 | RB_UC_DEV_PU_EN | RW | USB device enable and the internal pull-up resistor control bit in USB device mode, if it is 1, enable the USB device transmission and enable the internal pull-up resistor. | 0 |
| [5:4] | MASK_UC_SYS_CTRL | RW | See the table below to configure the USB system. | 00b |
| 3 | RB_UC_INT_BUSY | RW | Automatic pause enable bit before USB transfer completion interrupt flag is not reset.<br>1: Automatic pause before the interrupt flag UIF_TRANSFER is reset. In the device mode, the busy NAK is automatically answered; in the host mode, the subsequent transmission is automatically suspended;<br>0: No pause. | 0 |
| 2 | RB_UC_RESET_SIE | RW | USB protocol processor software reset control bit:<br>1: Force reset USB protocol processor (SIE). Need software to reset;<br>0: No reset. | 1 |
| 1 | RB_UC_CLR_ALL | RW | Reset USB's FIFO and interrupt flag:<br>1: Force emptying and clearing;<br>0: No clearing. | 1 |
| 0 | RB_UC_DMA_EN | RW | USB's DMA and DMA interrupt control bit:<br>1: Enable DMA feature and DMA interrupt;<br>0: Turn off DMA. | 0 |

The USB system control combination consists of RB_UC_HOST_MODE and MASK_UC_SYS_CTRL:

| RB_UC_HOST_MODE | MASK_UC_SYS_CTRL | Description of USB system control |
|---|---|---|
| 0 | 00 | Disable the USB device function and turn off the internal pull-up resistor. |
| 0 | 01 | Enable the USB device function, turn off the internal pull-up resistor, and add an external pull-up. |
| 0 | 1x | Enable the USB device function and enable the internal 1.5K pull-up resistor. This pull-up resistor takes precedence over the pull-down resistor and can also be used in GPIO mode. |
| 1 | 00 | USB host mode, in normal working status. |

| | | |
|---|---|---|
| 1 | 01 | USB host mode, force DP/DM to output SE0 status. |
| 1 | 10 | USB host mode, force DP/DM to output J status. |
| 1 | 11 | USB host mode, force DP/DM to output K status/wake up. |

**USB Interrupt Enable Register (R8_USB_INT_EN)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | RB_UIE_DEV_SOF | RW | In USB device mode, receive SOF packet interrupt:<br>1: Enable interrupt; 0: Disable interrupt. | 0 |
| 6 | RB_UIE_DEV_NAK | RW | USB device mode, receive NAK interrupt:<br>1: Enable interrupt; 0: Disable interrupt. | 0 |
| 5 | Reserved | RO | Reserved | 0 |
| 4 | RB_UIE_FIFO_OV | RW | FIFO overflow interrupt:<br>1: Enable interrupt; 0: Disable interrupt. | 0 |
| 3 | RB_UIE_HST_SOF | RW | In USB host mode, SOF timing interrupt:<br>1: Enable interrupt; 0: Disable interrupt. | 0 |
| 2 | RB_UIE_SUSPEND | RW | USB bus suspend or wake event interrupt:<br>1: Enable interrupt; 0: Disable interrupt. | 0 |
| 1 | RB_UIE_TRANSFER | RW | USB transfer completion interrupt:<br>1: Enable interrupt; 0: Disable interrupt. | 0 |
| 0 | RB_UIE_DETECT | RW | In USB host mode, USB device connection or disconnection event interrupt:<br>1: Enable interrupt; 0: Disable interrupt. | 0 |
| | RB_UIE_BUS_RST | RW | In USB device mode, USB bus reset event interrupt:<br>1: Enable interrupt; 0: Disable interrupt. | 0 |

**USB Device Address Register (R8_USB_DEV_AD)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | RB_UDA_GP_BIT | RW | USB universal flag bit, user-defined. | 0 |
| [6:0] | MASK_USB_ADDR | RW | Host mode: The address of USB device currently operating;<br>Device mode: The USB's own address. | 0000000b |

**USB Miscellaneous Status Register (R8_USB_MIS_ST)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | RB_UMS_SOF_PRES | RO | The SOF packet indication status bit in the USB host mode:<br>1: The SOF packet will be sent. If there are other USB data packets, it will be automatically delayed:<br>0: No SOF packet is sent. | X |
| 6 | RB_UMS_SOF_ACT | RO | SOF packet transmission status bit in the USB host mode:<br>1: SOF packet is being sent;<br>0: Transmission is complete or idle. | X |
| 5 | RB_UMS_SIE_FREE | RO | Idle status bit of the USB protocol processor:<br>1: Protocol is idle:<br>0: Busy. USB transfer is in progress. | 1 |
| 4 | RB_UMS_R_FIFO_RDY | RO | USB received FIFO data ready status bit:<br>1: Received FIFO is not empty:<br>O: Received FIFO is empty. | 0 |

| | | | | |
|---|---|---|---|---|
| 3 | RB_UMS_BUS_RESET | RO | USB bus reset status bit:<br>1: The current USB bus is in the reset state:<br>0: The current USB bus is in the non-reset state. | X |
| 2 | RB_UMS_SUSPEND | RO | USB suspend status bit:<br>1: The USB bus is in a suspended state and there is no USB activity for a while:<br>0: The USB bus is in a non-suspended state. | 0 |
| 1 | RB_UMS_DM_LEVEL | RO | In USB host mode, the device just connected to the USB port is in the level state of the DM pins and is used to determine the speed:<br>1: High level / low speed;<br>0: Low level / full speed. | 0 |
| 0 | RB_UMS_DEV_ATTACH | RO | USB device connection status bit for ports in USB host mode:<br>1: The port is connected to a USB device;<br>0: The port has no USB device connection. | 0 |

**USB Interrupt Flag Register (R8_USB_INT_FG)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | RB_U_IS_NAK | RO | In USB device mode, the NAK response status bit:<br>1: Respond to NAK during the current USB transfer process;<br>0: No NAK response. | 0 |
| 6 | RB_U_TOG_OK | RO | Current USB transfer DATA0/1 sync flag match status bit:<br>1: Sync; 0: Not synchronized. | 0 |
| 5 | RB_U_SIE_FREE | RO | USB protocol processor idle status bit:<br>1: USB is idle;<br>0: Busy, USB transfer is in progress. | 1 |
| 4 | RB_UIF_FIFO_OV | RW | USB FIFO overflow interrupt flag bit. Write 1 to reset:<br>1: FIFO overflow trigger;<br>0: No event. | 0 |
| 3 | RB_UIF_HST_SOF | RW | SOF timer interrupt flag bit in USB host mode. Write 1 to reset:<br>1: SOF packet transmission completion trigger;<br>0: No event. | 0 |
| 2 | RB_UIF_SUSPEND | RW | USB bus suspend or wake event interrupt flag bit. Write 1 to reset:<br>1: USB suspend event or wake-up event trigger;<br>0: No event. | 0 |
| 1 | RB_UIF_TRANSFER | RW | USB transfer completion interrupt flag bit. Write 1 to reset:<br>1: A USB transfer completion trigger;<br>0: No event. | 0 |
| 0 | RB_UIF_DETECT | RW | USB device connection or disconnection event interrupt flag bit in the USB host mode. Write 1 to reset:<br>1: Detected USB device connection or disconnection trigger;<br>0: No event. | 0 |
| | RB_UIF_BUS_RST | RW | USB bus reset event interrupt flag bit in USB device mode. Write 1 to reset:<br>1: USB bus reset event trigger;<br>0: No event. | 0 |

**USB Interrupt Status Register (R8_USB_INT_ST)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | RB_UIS_IS_NAK | RO | In USB device mode, the NAK response status bit is the same as RB_U_ IS_ NAK:<br>1: Respond to NAK during current USB transfer;<br>0: No NAK response. | 0 |
| 6 | RB_UIS_TOG_OK | RO | The current USB transfer DATA0/1 sync flag matches the status bit, the same as RB_U_TOG_OK:<br>1: Sync; 0: Not synchronized. | 0 |
| [5:4] | MASK_UIS_TOKEN | RO | In the device mode, the token PID ID of the current USB transfer transaction. | XXb |
| [3:0] | MASK_UIS_ENDP | RO | In the device mode, the endpoint number of the current USB transfer transaction. | XXXXb |
| | MASK_UIS_H_RES | RO | The response PID ID of the current USB transfer transaction in the host mode: 0000 indicates that the device has no response or timeout; other values indicate the response to PID. | XXXXb |

MASK_UIS_TOKEN is used for identifying the token PID of the current USB transfer transaction in USB device mode: 00 for OUT packets; 01 for SOF packets; 10 for IN packets; 11 for SETUP packets.
MASK_UIS_H_RES is only valid in the host mode. In the host mode, if the host sends an OUT/SETUP token packet, the PID is the handshake packet ACK/NAK/STALL, or the device has no response/timeout. If the host sends an IN token packet, the PID is the PID of the data packet (DATA0/DATA1) or the handshake packet PID.

**USB Receive Length Register (R8_USB_RX_LEN)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7:0] | R8_USB_RX_LEN | RO | The number of data bytes received by the current USB endpoint. | XXh |

### 17.2.2 Device register

In the USB device mode, a chip is provided with five sets of bidirectional endpoints: Endpoint 0, Endpoint 1, Endpoint 2, Endpoint 3 and Endpoint 4. The maximum data packet length is 64 bytes for all endpoints.
Endpoint 0 is the default endpoint and supports control transfers. One 64-byte data buffer zone is shared by transmission and reception. Endpoint 1, Endpoint 2 and Endpoint 3 each include a transmit endpoint IN and a receive endpoint OUT. Both transmission and reception have an independent 64-byte or dual 64-byte data buffer zone that supports bulk transfer, interrupt transfer, and real-time/synchronization transfer.
Endpoint 4 includes a transmit endpoint IN and a receive endpoint OUT. Both transmission and reception are provided with an independent 64-byte data buffer zone that supports bulk transfers, interrupt transfers, and real-time/synchronous transfers.
Each set of endpoints has a control register R8_UEPn_CTRL and a transmit length register R8_UEPn_T_LEN (n=0/1/2/3/4) that is used to set the synchronization trigger bit of the endpoint, the response to the OUT transaction and the IN transaction, the length of transmit data, etc.
Whether to enable the USB bus pull-up resistor which is necessary for a USB device can be set by software at any time. When RB_UC_DEV_PU_EN in the USB control register R8_USB_CTRL is set to 1, the controller is set according to the speed of RB_UD_LOW_SPEED. Internally, pull-up resistors are connected with the DP/DM pins of the USB bus, and the function of USB device is enabled.
When a USB bus reset, a USB bus suspend or a wake-up event is detected, or when USB successfully finishes processing data transmission or data reception, the USB protocol processor will set the corresponding interrupt flag. If an interrupt is enabled, the corresponding interrupt request will also be generated. The application can be used to directly query or analyze the interrupt flag register R8_USB_INT_FG in the USB interrupt service program. Corresponding processing is made according to RB_UIF_BUS_RST and RB_UIF_SUSPEND; if RB_UIF_TRANSFER is valid, it is necessary to continue analyzing the USB interrupt status register R8_USB_INT_ST and correspondingly process according to the current endpoint number of MASK_UIS_ENDP and the current transaction token PID ID of MASK_UIS_TOKEN. If the synchronization trigger bit

RB_UEP_R_TOG of the OUT transaction of each endpoint is set in advance, RB_U_TOG_OK or RB_UIS_TOG_OK can be used to judge whether the synchronization trigger bit of the currently received data packet matches the synchronization trigger bit of the endpoint. If the data is synchronized, the data is valid; if not, the data should be discarded. After each processing of the USB transmission or reception interrupt, the synchronization trigger bit of the corresponding endpoint should be correctly modified to detect whether the next transmitted data packet or the next received data packet is synchronous. In addition, RB_UEP_AUTO_TOG can be set to implement the automatically flip of the corresponding synchronization trigger bits after the transmission is successful or the reception is successful.
The data that each endpoint is ready to send is in its own buffer zone. The length of the data to be sent is independently set in R8_UEPn_T_LEN; the data received by each endpoint is in its own buffer zone, but the length of the received data is in the USB Receive Length Register R8_USB_RX_LEN, which can be distinguished according to the current endpoint number when the USB receives an interrupt.

Table 17-2 List of USB device related registers
(those marked as gray are reset and controlled by RB_UC_RESET_SIE)

| Name | Access address | Description | Reset value |
|---|---|---|---|
| R8_ UDEV_ CTRL | 0x40008001 | USB Device Physical Port Control Register | 0xX0 |
| R8_ UEP4_ 1_MOD | 0x4000800c | Endpoint 1/4 Mode Control Register | 0x00 |
| R8_ UEP2_ 3_MOD | 0x4000800d | Endpoint 2/3 Mode Control Register | 0x00 |
| R16_ UEP0_ DMA | 0x40008010 | Endpoint 0 Buffer Zone Start Address | 0xXXXX |
| R16_ UEP1_ DMA | 0x40008014 | Endpoint 1 Buffer Zone Start Address | 0xXXXX |
| R16_ UEP2_ DMA | 0x40008018 | Endpoint 2 Buffer Zone Start Address | 0xXXXX |
| R16_ UEP3_ DMA | 0x4000801c | Endpoint 3 Buffer Zone Start Address | 0xXXXX |
| R8_ UEP0_ T_ LEN | 0x40008020 | Endpoint 0 Transmit Length Register | 0xXX |
| R8_ UEP0_ CTRL | 0x40008022 | Endpoint 0 Control Register | 0x00 |
| R8_ UEP1_ T_ LEN | 0x40008024 | Endpoint 1 Transmit Length Register | 0xXX |
| R8_ UEP1_ CTRL | 0x40008026 | Endpoint 1 Control Register | 0x00 |
| R8_ UEP2_ T_LEN | 0x40008028 | Endpoint 2 Transmit Length Register | 0xXX |
| R8_ UEP2_ CTRL | 0x4000802a | Endpoint 2 Control Register | 0x00 |
| R8_ UEP3_ T_LEN | 0x4000802c | Endpoint 3 Transmit Length Register | 0xXX |
| R8_ UEP3_ CTRL | 0x4000802e | Endpoint 3 Control Register | 0x00 |
| R8_ UEP4_ T_LEN | 0x40008030 | Endpoint 4 Transmit Length Register | 0xXX |
| R8_UEP4_CTRL | 0x40008032 | Endpoint 4 Control Register | 0x00 |

**USB Device Physical Port Control Register (R8_UDEV_CTRL)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | RB_UD_PD_DIS | RW | Internal pull-down resistor control bit of USB device port UD+/UD- pins:<br>1: Disable internal pull-down;<br>0: Enable internal pull-down.<br>It can be used to provide pull-down resistors in GPIO mode. | 1 |
| 6 | Reserved | RO | Reserved | 0 |
| 5 | RB_UD_DP_PIN | RO | Current UD+ pin status:<br>1: High level; 0: Low level. | X |

| | | | | |
|---|---|---|---|---|
| 4 | RB_UD_DM_PIN | RO | Current UD-pin status:<br>1: High level; 0: Low level. | X |
| 3 | Reserved | RO | Reserved | 0 |
| 2 | RB_UD_LOW_SPEED | RW | USB device physical port low speed mode enable bit:<br>1: Select 1.5Mbps low speed mode;<br>0: Select 12Mbps full speed mode. | 0 |
| 1 | RB_UD_GP_BIT | RW | USB device mode universal flag bit, user-defined. | 0 |
| 0 | RB_UD_PORT_EN | RW | USB device physical port enable bit:<br>1: Enable physical port; 0: Disable physical port. | 0 |

**Endpoint 1/4 Mode Control Register (R8_UEP4_1_MOD)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | RB_UEP1_RX_EN | RW | 1: Enable Endpoint 1 reception (OUT);<br>0: Disable Endpoint 1 reception. | 0 |
| 6 | RB_UEP1_TX_EN | RW | 1: Enable Endpoint 1 transmission (IN);<br>0: Disable Endpoint 1 transmission. | 0 |
| 5 | Reserved | RO | Reserved | 0 |
| 4 | RB_UEP1_BUF_MOD | RW | Endpoint 1 data buffer zone mode control bit. | 0 |
| 3 | RB_UEP4_RX_EN | RW | 1: Enable Endpoint 4 reception (OUT);<br>0: Disable Endpoint 4 reception. | 0 |
| 2 | RB_UEP4_TX_EN | RW | 1: Enable Endpoint 4 transmission (IN);<br>0: Disable Endpoint 4 transmission. | 0 |
| [1:0] | Reserved | RO | Reserved | 0 |

The combination of bUEP4_RX_EN and bUEP4_TX_EN configure the data buffer modes of USB Endpoint 0 and Endpoint 4. Please refer to the following table:

Table 17-3 Buffer zone modes of Endpoint 0 and Endpoint 4

| bUEP4_RX_EN | bUEP4_TX_EN | Description: Arranged from low to high with UEP0_DMA as the start address |
|---|---|---|
| 0 | 0 | Endpoint 0 single 64-byte transceiver shared buffer zone (IN and OUT). |
| 1 | 0 | Endpoint 0 single 64-byte transceiver shared buffer zone: Endpoint 4 single 64-byte reception buffer zone (OUT). |
| 0 | 1 | Endpoint 0 single 64-byte transceiver shared buffer zone: Endpoint 4 single 64-byte transmit buffer zone (IN). |
| 1 | 1 | Endpoint 0 single 64-byte transceiver shared buffer zone: Endpoint 4 single 64-byte reception buffer zone (OUT);<br>Endpoint 4 single 64-byte reception buffer zone (IN). A total of 192 bytes are arranged as follows:<br>UEPO_DMA+0 Address: 64-byte start address of Endpoint 0 Transceiver Shared Buffer Zone;<br>UEPO_DMA+64 address: 64-byte start address of Endpoint 4 Receive Buffer Zone;<br>UEPO_DMA+128 address: 64-byte start address of Endpoint 4 Transmit Buffer Zone. |

**Endpoint 2/3 Mode Control Register (R8_UEP2_3_MOD)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | RB_UEP3_RX_EN | RW | 1: Enable Endpoint 3 reception (OUT);<br>0: Disable Endpoint 3 reception. | 0 |
| 6 | RB_UEP3_TX_EN | RW | 1: Enable Endpoint 3 transmission (IN);<br>0: Disable Endpoint 3 transmission. | 0 |
| 5 | Reserved | RO | Reserved | 0 |
| 4 | RB_UEP3_BUF_MOD | RW | Endpoint 3 data buffer zone mode control bit. | 0 |
| 3 | RB_UEP2_RX_EN | RW | 1: Enable Endpoint 2 reception (OUT);<br>0: Disable Endpoint 2 reception. | 0 |
| 2 | RB_UEP2_TX_EN | RW | 1: Enable Endpoint 2 transmission (IN);<br>0: Disable Endpoint 2 transmission. | 0 |
| 1 | Reserved | RO | Reserved | 0 |
| 0 | RB_UEP2_BUF_MOD | RW | Endpoint 2 data buffer zone mode control bit. | 0 |

The data buffer zone modes of USB Endpoint 1, Endpoint 2 and Endpoint 3 are respectively configured by the combination of RB_UEPn_RX_EN and RB_UEPn_TX_EN and RB_UEPn_BUF_MOD (n=1/2/3). For details, please refer to the following table. Specifically, in the mode of dual 64-byte buffer zone, the first 64-byte buffer zone is selected according to RB_UEP_*_TOG=0 at time of USB data transmission, and the last 64-byte buffer zone is selected according to RB_UEP_*_TOG=1. RB_UEP_AUTO_TOG=1 can be set to be automatic switching.

Table 17-4 Buffer zone modes of Endpoint n (n=1/2/3)

| RB_UEPn_RX_EN | RB_UEPn_TX_EN | RB_UEPn_BUF_MOD | Description: Arranged from low to high with R16_UEPn_DMA as the start address |
|---|---|---|---|
| 0 | 0 | X | The endpoint is disabled and the R16_UEPn_ DMA buffer zone is not used. |
| 1 | 0 | 0 | Single 64-byte reception buffer zone (OUT). |
| 1 | 0 | 1 | Dual 64-byte reception buffer zone (OUT), selected by RB_UEP_R_TOG. |
| 0 | 1 | 0 | Single 64-byte transmission buffer zone (N). |
| 0 | 1 | 1 | Dual 64-byte transmission buffer zone (IN), selected by RB_UEP_T_TOG. |
| 1 | 1 | 0 | Single 64-byte reception buffer zone (OUT), single 64-byte transmission buffer zone (IN). |
| 1 | 1 | 1 | Dual 64-byte reception buffer zone (OUT), selected by RB_UEP_R_TOG,<br>Dual 64-byte transmit buffer zone (IN), selected by RB_UEP_T_TOG.<br>All 256 bytes are arranged as follows:<br>UEPn_DMA+0 address: the endpoint reception address when RB_UEP_ R_TOG=0;<br>UEPn_DMA+64 address: the endpoint reception address when RB_UEP_ R_ TOG=1;<br>UEPn_DMA+128 address: the endpoint transmission address when RB_UEP_T_TOG=0;<br>UEPn_DMA+192 address: the endpoint transmission address when RB_UEP_ T_ TOG=1. |

**Endpoint n Buffer Zone Start Address (R16_UEPn_DMA) (n=0/1/2/3)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:0] | R16_UEPn_DMA | RW | Start Address of Endpoint n Buffer Zone<br>The low 15 bits are valid and the address must be 4 bytes aligned. | XXXXh |

Note: The length of the buffer zone that receives the data >= min (the maximum length of data packet that can be received + 2 bytes, 64 bytes).

**Endpoint n Transmit Length Register (R8_UEPn_T_LEN) (n=0/1/2/3/4)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7:0] | R8_UEPn_T_LEN | RW | Set the number of data bytes that the USB Endpoint n is ready to send. | XXh |

**Endpoint n Control Register (R8_UEPn_CTRL) (n=0/1/2/3/4)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | RB_UEP_R_TOG | RW | The desired trigger bit of the receiver of USB Endpoint n (processing the OUT transaction):<br>1: Expect DATA1; 0: Expect DATA0. | 0 |
| 6 | RB_UEP_T_TOG | RW | The prepared sync trigger bit of the transmitter of USB Endpoint n (processing the IN transaction):<br>1: Send DATA1; 0: Send DATA0. | 0 |
| 5 | Reserved | RO | Reserved | 0 |
| 4 | RB_UEP_AUTO_TOG | RW | Synchronous trigger bit auto flip enable control bit:<br>1: The corresponding synchronous trigger bit is automatically flipped after the data is sent or received successfully;<br>0: No automatic flip. Manual switch is allowed.<br>Only support Endpoint 1/2/3. | 0 |
| [3:2] | MASK_UEP_R_RES | RW | The response control of the receiver of Endpoint n to the OUT transaction:<br>00: Answer to ACK;<br>01: Timeout/no response, used for real-time/synchronous transmission of endpoints other than Endpoint 0;<br>10: Answer to NAK or busy;<br>11: Answer to STALL or error. | 00b |
| [1:0] | MASK_UEP_T_RES | RW | The response control of the transmitter of Endpoint n to the IN transaction:<br>00: DATA0/DATA1 data is ready and expect ACK;<br>01: Answer to DATA0/DATA1, expect no response, and used for real-time/synchronous transmission of endpoints other than Endpoint 0;<br>10: Answer to NAK or busy;<br>11: Answer to STALL or error. | 00b |

### 17.2.3 Host register

In the USB host mode, a chip is provided with a set of bidirectional host endpoints, including a transmit endpoint OUT and a receive endpoint IN. The maximum length of the data packet is 64 bytes. It supports control transfers, interrupt transfers, bulk transfers, and real-time/synchronous transfers.

Each USB transaction initiated by the host endpoint always sets the RB_UIF_TRANSFER interrupt flag

automatically after processing. The application can directly query or query and analyze the interrupt flag register R8_USB_INT_FG in the USB interrupt service program, and perform corresponding processing according to each interrupt flag; if RB_UIF_TRANSFER is valid, it is needed to continue analyzing the USB interrupt status register R8_USB_INT_ST and perform corresponding processing according to the response PID identification MASK_UIS_H_RES of the current USB transfer transaction.
If the synchronization trigger bit RB_UH_R_TOG of the IN transaction of the host receiving endpoint is set in advance, whether the synchronization trigger bit of the currently received data packet matches the synchronization trigger bit of the host receiving endpoint can be judged by RB_U_TOG_OK or RB_UIS_TOG_OK. If the data is synchronized, the data is valid; if not, the data should be discarded. After each processing of USB transmission or reception interrupt, the synchronization trigger bit of the corresponding host endpoint should be correctly modified to synchronize the next transmitted data packet and detect whether the next received data packet is synchronized. In addition, the setting of RB_UH_T_AUTO_TOG and RB_UH_R_AUTO_TOG can automatically flip the corresponding synchronization trigger bit after successful transmission or successful reception.
The USB host token setting register R8_UH_EP_PID is used to set the endpoint number of the target device being operated and the token PID packet identifier of this USB transfer transaction. The data corresponding to the SETUP token and the OUT token is provided by the host transmit endpoint. The data to be sent is in the R16_UH_TX_DMA buffer zone, and the length of the data to be sent is set in R16_UH_TX_LEN; the data corresponding to the IN token is returned to the host receive endpoint by the target device. The received data is stored in the R16_UH_RX_DMA buffer zone. The length of the received data is stored in R8_USB_RX_LEN.

Table 17-5 List of registers related to USB hosts
(Those marked as gray are reset and controlled by RB_UC_RESET_SIE)

| Name | Access address | Description | Reset value |
|---|---|---|---|
| R8_UHOST_CTRL | 0x40008001 | USB Host Physical Port Control Register | 0xX0 |
| R8_UH_EP_MOD | 0x4000800d | USB Host Endpoint Mode Control Register | 0x00 |
| R16_UH_RX_DMA | 0x40008018 | USB Host Receive Buffer Zone Start Address | 0xXXXX |
| R16_UH_TX_DMA | 0x4000801c | USB Host Transmit Buffer Zone Start Address | 0xXXXX |
| R8_UH_SETUP | 0x40008026 | USB Host Auxiliary Setting Register | 0x00 |
| R8_UH_EP_PID | 0x40008028 | USB Host Token Setting Register | 0x00 |
| R8_UH_RX_CTRL | 0x4000802a | USB Host Receive Endpoint Control Register | 0x00 |
| R8_UH_TX_LEN | 0x4000802c | USB Host Transmit Length Register | 0xXX |
| R8_UH_TX_CTRL | 0x4000802e | USB Host Transmit Endpoint Control Register | 0x00 |

**USB Host Physical Port Control Register (R8_UHOST_CTRL)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | RB_UH_PD_DIS | RW | Internal pull-down resistor control bit of USB host port UD+/UD- pins:<br>1: Disable internal pull-down;<br>0: Enable internal pull-down.<br>Can be used to provide pull-down resistors in the GPIO mode. | 1 |
| 6 | Reserved | RO | Reserved | 0 |
| 5 | RB_UH_DP_PIN | RO | Current UD+ pin status:<br>1: High level; 0: Low level. | X |
| 4 | RB_UH_DM_PIN | RO | Current UD-pin status:<br>1: High level; 0: Low level. | X |
| 3 | Reserved | RO | Reserved | 0 |
| 2 | RB_UH_LOW_SPEED | RW | USB host port low speed mode enable bit:<br>1: Select 1.5Mbps low speed mode;<br>0: Select 12Mbps full speed mode. | 0 |

| | | | | |
|---|---|---|---|---|
| 1 | RB_UH_BUS_RESET | RW | USB host mode bus reset control bit:<br>1: Force to output USB bus reset;<br>0: End output. | 0 |
| 0 | RB_UH_PORT_EN | RW | USB host port enable bit:<br>1: Enable host port; 0: Disable host port.<br>This bit is automatically reset to zero when the USB device is disconnected. | 0 |

**USB Host Endpoint Mode Control Register (R8_UH_EP_MOD)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | Reserved | RO | Reserved | 0 |
| 6 | RB_UH_EP_TX_EN | RW | Host transmit endpoint transmission (SETUP/OUT) enable bit:<br>1: Enable endpoint transmission;<br>0: Disable endpoint transmission. | 0 |
| 5 | Reserved | RO | Reserved | 0 |
| 4 | RB_UH_EP_TBUF_MOD | RW | The mode of the host transmit endpoint transmitting data to the buffer zone control bit. | 0 |
| 3 | RB_UH_EP_RX_EN | RW | Host receive endpoint reception (IN) enable bit:<br>1: Enable endpoint reception;<br>0: Disable endpoint reception. | 0 |
| [2:1] | Reserved | RO | Reserved | 00b |
| 0 | RB_UH_EP_RBUF_MOD | RW | The mode of the host receive endpoint receiving data to the buffer zone control bit. | 0 |

The mode of the host transmit endpoint data buffer zone is controlled by the combination of RB_UH_EP_TX_EN and RB_UH_EP_TBUF_MOD. Please refer to the following table.

Table 17-6 Host transmit buffer zone mode

| RB_UH_EP_TX_EN | RB_UH_EP_TBUF_MOD | Description: R16_UH_TX_DMA as the start address |
|---|---|---|
| 0 | X | The endpoint is disabled and the R16_UH_TX_DMA buffer zone is not used. |
| 1 | 0 | Single 64-byte transmit buffer zone (SETUP/OUT). |
| 1 | 1 | Dual 64-byte transmit buffer zone, selected by RB_UH_T_TOG:<br>When RB_UH_T_TOG=0, select the first 64-byte buffer zone;<br>When RB_UH_T_TOG=1, select the last 64-byte buffer zone. |

The mode of the host receive endpoint data buffer zone is controlled by the combination of RB_UH_EP_RX_EN and RB_UH_EP_RBUF_MOD. Please refer to the following table.

Table 17-7 Host receive buffer zone mode

| RB_UH_EP_RX_EN | RB_UH_EP_RBUF_MOD | Structure description: R16_UH_TX_DMA as the start address |
|---|---|---|
| 0 | X | The endpoint is disabled and the R16_UH_RX_DMA buffer zone is not used. |
| 1 | 0 | Single 64-byte receive buffer zone (IN). |

| | | |
|---|---|---|
| 1 | 1 | Dual 64-byte transmit buffer zone, selected by RB_UH_R_TOG:<br>When RB_UH_R_TOG=0, select the first 64-byte buffer zone;<br>When RB_UH_R_TOG=1, select the last 64-byte buffer zone. |

**USB Host Receive Buffer Zone Start Address (R16_UH_RX_DMA)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:0] | R16_UH_RX_DMA | RW | Start address of the host endpoint data receive buffer zone. The low 15 bits are valid and the address must be 4 bytes aligned. | XXXXb |

**USB Host Transmit Buffer Zone Start Address (R16_UH_TX_DMA)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:0] | R16_UH_TX_DMA | RW | Start address of the host endpoint data transmit buffer zone. The low 15 bits are valid and the address must be 4 bytes aligned. | XXXXb |

**USB Host Auxiliary Setting Register (R8_UH_SETUP)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | RB_UH_PRE_PID_EN | RW | Low speed preamble packet PRE PID enable bit:<br>1: Enable to communicate with a low-speed USB device via an external HUB.<br>0: Disable the low-speed preamble packet. | 0 |
| 6 | RB_UH_SOF_EN | RW | Automatically generate SOF packet enable bit:<br>1: The host automatically generates a SOF packet:<br>0: Not generated automatically, but can be generated manually. | 0 |
| [5:0] | Reserved | RO | Reserved | 000000b |

**USB Host Token Setting Register (R8_UH_EP_PID)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7:4] | MASK_UH_TOKEN | RW | Set the token PID packet ID for this USB transfer transaction. | 0000b |
| [3:0] | MASK_UH_ENDP | RW | Set the endpoint number of the target device that is being operated this time. | 0000b |

**USB Host Receive Endpoint Control Register (R8_UH_RX_CTRL)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | RB_UH_R_TOG | RW | The expected sync trigger bit for the USB host receiver (handling the IN transaction):<br>1: Expected DATA1; 0: Expected DATA0. | 0 |
| [6:5] | Reserved | RO | Reserved | 00b |

| | | | | |
|---|---|---|---|---|
| 4 | RB_UH_R_AUTO_TOG | RW | Synchronous trigger bit auto flip enable control bit:<br>1: After data is successfully received, the corresponding expected sync trigger bit (RB_UH_R_TOG) is automatically flipped;<br>0: No automatic flip. Manual switch is allowed. | 0 |
| 3 | Reserved | RO | Reserved | 0 |
| 2 | RB_UH_R_RES | RW | Host receiver respond control bit to IN transaction:<br>1: No response, used for real-time/synchronous transfer of endpoints other than Endpoint 0;<br>0: Answer to ACK. | 0 |
| [1:0] | Reserved | RO | Reserved | 00b |

**USB Host Transmit Length Register (R8_UH_TX_LEN)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7:0] | R8_UH_TX_LEN | RW | Set the number of data bytes that the USB host transmit endpoint is ready to send. | XXh |

**USB Host Transmit Endpoint Control Register (R8_UH_TX_CTRL)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | Reserved | RO | Reserved | 0 |
| 6 | RB_UH_T_TOG | RW | The sync trigger bit prepared by the USB host transmitter (processing SETUP/OUT transaction):<br>1: indicates that DATA1 is sent;<br>0: indicates that DATA0 is sent. | 0 |
| 5 | Reserved | RO | Reserved | 0 |
| 4 | RB_UH_T_AUTO_TOG | RW | Synchronous trigger bit auto flip enable control bit:<br>1: After the data is sent successfully, the corresponding synchronization trigger bit (RB_UH_T_TOG) is automatically flipped;<br>0: No automatic flip. Manual switch is allowed. | 0 |
| [3:1] | Reserved | RO | Reserved | 000b |
| 0 | RB_UH_T_RES | RW | USB host transmitter response control bit to SETUP/OUT transaction:<br>1: Expect no response, used for real-time/synchronous transfers of endpoints other than endpoint 0;<br>0: Expect to answer to ACK. | 0 |

# Chapter 18 Ethernet Controller (ETH)

## 18.1 Ethernet controller introduction

A chip is integrated with the Ethernet controller MAC, PHY and DMA and is compatible with the IEEE802.3 protocol. The internal DMA transfers data and receives data to the system RAM. The PHY physical layer is a 10 Mbit/s Ethernet transceiver that provides a portion of the network PHY registers to set the performance of transmission and reception.

The underlying operations of the network mainly provide application support with the subroutine library. The registers are not described in detail. Main features are:

- Support full-duplex and half-duplex.
- Support the settings of short packet filling.
- Support CRC setting and filling.
- Support jumbo frame reception.
- Support the transmission and setting of pause frames.
- Support Auto-Negotiation mechanism.
- Support DMA for transmitting and receiving data.
- The PHY transceiver is compatible with 10BASE-T, and the sending module supports the energy saving mode.
- Built-in 50ohm transmission impedance matching resistor. External connection is optional.
- Provide a globally unique MAC address assigned by the IEEE.

## 18.2 Register description

Table 18-1 List of registers related to Ethernet controller

| Name | Offset address | Description | Reset value |
|---|---|---|---|
| R8_ETH_EIE | 0x40009003 | Interrupt Enable Register | 0x00 |
| R8_ETH_EIR | 0x40009004 | Interrupt Flag Register | 0x00 |
| R8_ETH_ESTAT | 0x40009005 | Status Register | 0x00 |
| R8_ETH_ECON2 | 0x40009006 | PHY Analog Parameter Setting Register | 0x06 |
| R8_ETH_ECON1 | 0x40009007 | Transceiver Control Register | 0x00 |
| R32_ETH_TX | 0x40009008 | Transmit DMA Control Register | 0xXXXXXXXX |
| R16_ETH_ETXST | 0x40009008 | Transmit DMA Buffer Start Address Register | 0xXXXX |
| R16_ETH_ETXLN | 0x4000900A | Transmit Length Register | 0xXXXX |
| R32_ETH_RX | 0x4000900C | Receive DMA Control Register | 0x00000000 |
| R16_ETH_ERXST | 0x4000900C | Receive DMA Buffer Start Address Register | 0x0000 |
| R16_ETH_ERXLN | 0x4000900E | Receive Length Register | 0x0000 |
| R8_ETH_MACON1 | 0x40009019 | Mac Layer Flow Control Register | 0x00 |
| R8_ETH_MACON2 | 0x4000901A | Mac Layer Packet Control Register | 0x00 |
| R8_ETH_MABBIPG | 0x4000901B | Minimum Inter-packet Interval Register | 0x10 |
| R32_ETH_TIM | 0x4000901C | Flow Control Pause Frame Time Register | 0xXXXXXXXX |
| R16_ETH_EPAUS | 0x4000901C | Flow Control Pause Frame Time Register | 0xXXXX |
| R16_ETH_MAMXFL | 0x4000901E | Maximum Receive Packet Length Register | 0x0000 |
| R16_ETH_MIRD | 0x40009020 | MII Read Data Register | 0x1100 |
| R32_ETH_MIWR | 0x40009024 | MII Write Register | 0x00000000 |
| R8_ETH_MIREGADR | 0x40009024 | MII Address Register | 0x00 |
| R8_ETH_MISTAT | 0x40009025 | MII Status Register | 0x00 |

| | | | |
|---|---|---|---|
| R16_ETH_MIWR | 0x40009026 | MII Write Data Register | 0x0000 |
| R32_ETH_MAADRL | 0x40009028 | MAC Address Low Byte Register | 0xXXXXXXXX |
| R16_ETH_MAADRH | 0x4000902C | MAC Address High Byte Register | 0xXXXX |

**Interrupt Enable Register (R8_ETH_EIE)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | RB_ETH_EIE_INTIE | RW | Ethernet interrupt enable:<br>0: Turn off interrupt; 1: Turn on interrupt. | 0 |
| 6 | RB_ETH_EIE_RXIE | RW | Reception completion interrupt enable:<br>0: Turn off interrupt; 1: Turn on interrupt. | 0 |
| 5 | Reserved | RO | Reserved | 0 |
| 4 | RB_ETH_EIE_LINKIE | RW | Link change interrupt enable:<br>0: Turn off interrupt; 1: Turn on interrupt. | 0 |
| 3 | RB_ETH_EIE_TXIE | RW | Transmission completion interrupt enable:<br>0: Turn off interrupt; 1: Turn on interrupt. | 0 |
| 2 | RB_ETH_EIE_R_EN50 | RW | Built-in 50 ohm impedance matching resistor enable:<br>0: On-chip resistor disconnected;<br>1: On-chip resistor connection. | 0 |
| 1 | RB_ETH_EIE_TXERIE | RW | Transmission error interrupt enable:<br>0: Turn off interrupt; 1: Turn on interrupt. | 0 |
| 0 | RB_ETH_EIE_RXERIE | RW | Reception error interrupt enable:<br>0: Turn off interrupt; 1: Turn on interrupt. | 0 |

**Interrupt Flag Register (R8_ETH_EIR)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | Reserved | RO | Reserved | 0 |
| 6 | RB_ETH_EIR_RXIF | RW1 | Reception completion flag. | 0 |
| 5 | Reserved | RO | Reserved | 0 |
| 4 | RB_ETH_EIR_LINKIF | RW1 | Link change flag. | 0 |
| 3 | RB_ETH_EIR_TXIF | RW1 | Transmission completion flag. | 0 |
| 2 | Reserved | RO | Reserved | 0 |
| 1 | RB_ETH_EIR_TXERIF | RW1 | Transmission error flag. | 0 |
| 0 | RB_ETH_EIR_RXERIF | RW1 | Reception error flag. | 0 |

**Status Register (R8_ETH_ESTAT)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | RB_ETH_ESTAT_INT | RW1 | Interrupted. | 0 |
| 6 | RB_ETH_ESTAT_BUFER | RW1 | Buffer error. | 0 |
| 5 | RB_ETH_ESTAT_RXCRCER | RO | Receive CRC error. | 0 |
| 4 | RB_ETH_ESTAT_RXNIBBLE | RO | Receive nibble error. | 0 |
| 3 | RB_ETH_ESTAT_RXMORE | RO | Reception exceeds the set maximum packet. | 0 |
| 2 | RB_ETH_ESTAT_RXBUSY | RO | Reception is in progress. | 0 |
| 1 | RB_ETH_ESTAT_TXABRT | RO | Transmission is interrupted by the MCU. | 0 |
| 0 | Reserved | RO | Reserved | 0 |

**PHY Analog Parameter Setting Register (R8_ETH_ECON2)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7:4] | Reserved | RO | Reserved | 0000b |
| [3:1] | RB_ETH_ECON2_RX<br>RB_ETH_ECON2_MUST | RW | Reserved. 011 must be written. | 011b |
| 0 | RB_ETH_ECON2_TX | RW | Transmitting end energy-saving drive control:<br>0: Rated drive;<br>1: Energy-saving drive. | 0 |

**Transceiver Control Register (R8_ETH_ECON1)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | RB_ETH_ECON1_TXRST | RW | Transmission module reset:<br>0: No reset;<br>1: Reset the transmission module. | 0 |
| 6 | RB_ETH_ECON1_RXRST | RW | Reception module reset:<br>0: No reset;<br>1: Reset the reception module. | 0 |
| [5:4] | Reserved | RO | Reserved | 00b |
| 3 | RB_ETH_ECON1_TXRTS | RW | Transmission starts and is automatically reset to zero after the transmission is completed:<br>1: Start transmission; 0: No action. | 0 |
| 2 | RB_ETH_ECON1_RXEN | RW | Reception enable control:<br>0: Turn off reception; 1: Turn on reception. | 0 |
| [1:0] | Reserved | RO | Reserved | 00b |

**Transmit DMA Buffer Address Register (R16_ETH_ETXST)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:0] | R16_ETH_ETXST | RW | Transmit the start address of the DMA buffer zone. The low 15 bits are valid and the address must be 4 bytes aligned. | XXXXh |

**Transmit Length (R16_ETH_ETXLN)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:0] | R16_ETH_ETXLN | RW | The length of transmission | XXXXh |

**Receive DMA Buffer Address Register (R16_ETH_ERXST)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:0] | R16_ETH_ERXST | RW | Receive the start address of the DMA buffer zone. The low 15 bits are valid and the address must be 4 bytes aligned. | XXXXh |

**Receive Length Register (R16_ETH_ERXLN)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:0] | R16_ETH_ERXLN | RO | The length of reception | 0000h |

**Mac Layer Flow Control Register (R8_ETH_MACON1)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7:6] | Reserved | RO | Reserved | 00b |
| [5:4] | RB_ETH_MACON1_FCEN | RW | Pause frame is set to be valid under full duplex:<br>00: Stop sending a pause frame;<br>01: Send a pause frame and then stop sending;<br>10: Periodically send a pause frame;<br>11: Send 0 timer pause frame and then stop sending. | 00b |
| 3 | RB_ETH_MACON1_TXPAUS | RW | Transmit pause frame enable control:<br>0: No pause frame is transmitted;<br>1: Enable transmission. | |
| 2 | RB_ETH_MACON1_RXPAUS | RW | Receive pause frame enable:<br>0: No pause frame is received;<br>1: Enable reception. | 0 |
| 1 | RB_ETH_MACON1_PASSALL | RW | Control frame settings:<br>0: Control frames will be filtered;<br>1: Control frames that are not filtered will be written to the cache. | 0 |
| 0 | RB_ETH_MACON1_MARXEN | RW | MAC layer receive enable:<br>0: MAC does not receive data;<br>1: Enable MAC to receive. | 0 |

**Mac Layer Packet Control Register (R8_ETH_MACON2)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7:5] | RB_ETH_MACON2_PADCFG | RW | Set the filling of short packets:<br>7: All short packets are filled with 0 to 64 bytes, and then 4 bytes of CRC;<br>6: Short packets are not filled;<br>5: It is detected that the VLAN network packet with the field of 8100h is automatically filled with 0 to 64 bytes, otherwise the short packet is filled with 60 bytes of 0 and then 4 bytes of CRC after filling;<br>4: Short packets are not filled;<br>3: All short packets are filled with 0 to 64 bytes, then 4 bytes of CRC;<br>2: Short packets are not filled;<br>1: All short packets are filled with 0 to 60 bytes, then 4 bytes of CRC;<br>0: Short packets are not filled. | 000b |
| 4 | RB_ETH_MACON2_TXCRCEN | RW | Transmit and add CRC Control:<br>0: Hardware is not filled with CRC;<br>1: Hardware is filled with CRC. | 0 |
| 3 | RB_ETH_MACON2_PHDREN | RW | Special 4 bytes do not participate in the CRC check. | 0 |
| 2 | RB_ETH_MACON2_HFRMEN | RW | Allow receiving jumbo frames:<br>0: Do not allow jumbo frames to be received:<br>1: Allow reception. | 0 |
| 1 | Reserved | RO | Reserved | 0 |
| 0 | RB_ETH_MACON2_FULDPX | RW | Mode of Ethernet communication:<br>0: Half duplex; 1: Full duplex. | 0 |

**Minimum Inter-packet Interval Register (R8_ETH_MABBIPG)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| 7 | Reserved | RO | Reserved | 0 |
| [6:0] | R8_ETH_MABBIPG | RW | The minimum number of bytes of inter-packet interval. | 0010000b |

**Flow Control Pause Frame Time Register (R16_ETH_EPAUS)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:0] | R16_ETH_EPAUS | RW | Flow control pause the frame time. | XXXXh |

**Maximum Receive Packet Length Register (R16_ETH_MAMXFL)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:0] | R16_ETH_MAMXFL | RW | Maximum length of the reception packet. | 0000h |

**MII Read Data Register (R16_ETH_MIRD)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:0] | R16_ETH_MIRD | RW | MII read data register. | 1100h |

**MII Address Register (R8_ETH_MIREGADR)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7:5] | Reserved | RO | Reserved | 000b |
| [4:0] | RB_ETH_MIREGADR_MIRDL | RW | The address of PHY register | 00000b |

**MII Status Register (R8_ETH_MISTAT)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [7:1] | Reserved | RO | Reserved | 0000000b |
| 0 | R8_ETH_MII_STA | RO | Operation status of MII register:<br>1: Write MII register;<br>0: Read MII register. | 0 |

**MII Write Data Register (R16_ETH_MIWR)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [15:0] | R16_ETH_MIWR | RW | MII write data register. | 0000h |

**MAC Address Register (R32_ETH_MAADRL、R16_ETH_MAADRH)**

| Bit | Name | Access | Description | Reset value |
|---|---|---|---|---|
| [31:0] | R32_ETH_MAADRL | RW | MAC Address Byte 1~4. | XXXXXXXXh |
| [15:0] | R16_ETH_MAADRH | RW | MAC Address Byte 5~6. | XXXXh |

## 18.3 Operation guide

1. Initialize

(1) Configure the safety register to enter the safe mode, open the clock and power of the Ethernet network;
(2) Turn on the corresponding interrupt, and optionally enable the impedance matching resistor;
(3) Configure CRC function, MAC address;

(4) Set the cache;
(5) Start receiving, and open interrupt.

2. Transmit data
(1) Write the data length of R16_ETH_ETXLN;
(2) Write the data address of R16_ETH_ETXST;
(3) Enable the RB_ETH_ECON1_TXRTS flag to start transmission.

3. Receive data
(1) Set the reception address in advance, and enable to receive;
(2) Use interrupt or query the status of reception completion;
(3) Read the reception length of R16_ETH_ERXLN;
(4) Update the reception address of R16_ETH_ERXST.

The specific application should be based on the Ethernet protocol stack library and refer to the provided network application examples.

# Chapter 19 Wireless Communication

## 19.1 Introduction

A chip is integrated with low-power 2.4-GHz wireless communication modules, including RF transceivers, baseband, link control, and antenna matching networks. Bluetooth Low Energy is supported. More than one hundred registers are provided internally to adjust parameters, control processes and states. This manual will not provide a detailed description of registers. Only some registers commonly used by the application layer are listed. The underlying operations of wireless communication mainly provide application support with the help of the subroutine library.

Main features:

- Integrate 2.4GHz RF transceiver, BaseBand and LLE link control.
- Support low-power Bluetooth (BLE), and comply with the specifications of Bluetooth Low Energy 4.2.
- Single-ended RF interface; there is no need to match external inductor and capacitor and filter networks; simplified board-level design.
- Reception sensitivity is -93dBm.
- Programmable transmission power from -20dBm to +3dBm; support dynamic adjustment.
- The wireless communication distance of the PCB onboard antenna is about 100 meters when power is transmitted at 0dBm.
- After the built-in DC-DC conversion is used, the current is more than 6mA when power is transmitted at 0dBm.
- Support AES encryption and decryption.
- Support DMA.
- Provide optimized protocol stack and application layer APIs to support networking.

## 19.2 LLE Module

The LLE module supports automatic transceiving mode and manual transceiving mode as well as 5 independent hardware timers. It is able to control the time point of any process of sending and receiving data.

### 19.2.1 Register description

Table 19-1 List of registers related to LLE module

| Name | Offset address | Description | Reset value |
|---|---|---|---|
| R32_ LLE_ CTRL_ CMD | 0x4000C200 | LLE command | 0x00000000 |
| R32_ LLE_ CTRL_ CFG | 0x4000C204 | LLE configuration | 0x00000F01 |
| R32_ LLE_ STATUS | 0x4000C208 | LLE status | 0x00000000 |
| R32_ LLE_ INT_ EN | 0x4000C20C | LLE interrupt enable | 0x00003F1F |
| R32_ LLE_ CTRL_ MOD | 0x4000C250 | LLE mode control | 0x00000011 |

### 19.2.2 Function and configuration

Initialization process:

(1) Set R32_LLE_CTRL_MOD Register, write 58H, configure as the LLE mode, and turn on the power to enable the feature of DMA and BB module.

(2) Set R32_LLE_CTRL_CFG Register, and configure LLE as the mode of transmit-receive data automatically or manually.

(3) Set R32_LLE_CTRL_CMD Register and select 4 working states: transmit, receive, stop, and turn off. The details are as follows:

CMD_RX Use: Set the current mode to receive. In manual mode, after reception is completed, it will be reset to 0 automatically. In automatic mode, the process of cyclic transmission and reception will be repeated and can be stopped by CMD_STOP or CMD_SHUT.

CMD_TX Use: Set the current mode to transmit. In manual mode, after transmission is completed, it will be reset to 0 automatically. In automatic mode, the process of cyclic transmission and reception will be repeated and can be stopped by CMD_STOP or CMD_SHUT.

CMD_STOP Use: In the automatic transmission mode, when CMD_STOP is set to 1 during the frame interval,

the CMD_TX status will be reset and redirected to LLE_IDLE, and the software cannot directly reset the CMD_TX status. In the automatic reception mode, when CMD_STOP is set to 1 during the frame interval, the CMD_TX status will be reset and redirected to LLE_IDLE, and the software cannot directly reset the CMD_RX status.
CMD_SHUT Use: When CMD_SHUT is set, CMD_TX/CMD_RX/CMD_STOP/CMD_SHUT will all be reset and the LLE status machine will be reset.

## 19.3 DMA Module

A controller has 2 sets of DMA, and each set of DMA has two channels. The two channels of DMA0 are used to transmit data and receive data respectively. The two channels of DMA2 are used in the automatic mode. In the automatic transmission mode, the address of transmitting DMA and the address of receiving DMA can be configured at the same time, so that reconfiguration is not required during the frame interval.

### 19.3.1 Register description

Table 19-2 List of registers related to DMA module

| Name | Offset address | Description | Reset value |
|---|---|---|---|
| R32_DMA0_CTRL_CFG | 0x4000C008 | DMA0 configuration register | 0x00002000 |
| R32_DMA2_CTRL_CFG | 0x4000C00C | DMA2 configuration register | 0x00002000 |
| R32_DMA0_TX_SRC | 0x4000C010 | DMA0 source address<br>Used to configure the start address of transmitting data | 0x00000000 |
| R32_DMA2_TX_SRC | 0x4000C014 | DMA2 source address, automatic receiving mode<br>Used to configure the start address of transmitting data | 0x00000000 |
| R32_DMA0_RX_DST | 0x4000C018 | DMA0 destination address<br>Used to configure the start address of receiving data | 0x00000000 |
| R32_DMA2_RX_DST | 0x4000C01C | DMA2 destination address, automatic transmission mode<br>Used to configure the start address of receiving data | 0x00000000 |

### 19.3.2 Function and description

Operation flow of transmission:
(1) Set R32_DMA0_TX_SRC Register and write the address of the transmitted data.
(2) Set R32_DMA0_CTRL_CFG Register and write the length of the transmitted data.
(3) Set IRQMASK of R32_DMA0_CTRL_CFG Register to 1.
(4) In the automatic mode, set R32_DMA2_RX_DST Register and write the address of the received data.
(5) In the automatic mode, set IRQMASK of R32_DMA2_CTRL_CFG Register to 1.

Operation flow of reception:
(1) Set R32_DMA0_RX_DST Register and write the address of the received data.
(2) Set IRQMASK of R32_DMA0_CTRL_CFG Register to 1.
(3) In the automatic mode, set R32_DMA2_TX_SRC and write the address of the transmitted data.
(4) In the automatic mode, set R32_DMA2_CTRL_CFG Register and write the length of the transmitted data.
(5) In the automatic mode, set IRQMASK of R32_DMA2_CTRL_CFG Register to 1.

## 19.4 BB Module

### 19.4.1 Register description

Table 19-3 List of registers related to BB module

| Name | Offset address | Description | Reset value |
|---|---|---|---|
| R32_BB_CTRL_CFG | 0x4000C100 | BB Configuration Register | 0x000822A7 |

| | | | |
|---|---|---|---|
| R32_BB_TXCRC_INIT | 0x4000C104 | Transmit CRC Initial Value Configuration Register | 0x0056BC9B |
| R32_BB_TXACCS_ADDR | 0x4000C108 | Transmit Access Address Configuration Register | 0x8E89BED6 |
| R32_BB_RXCRC_INIT | 0x4000C10C | Receive CRC Initial Value Configuration Register | 0x0056BC9B |
| R32_BB_RXACCS_ADDR | 0x4000C110 | Receive Access Address Configuration Register | 0xBE89BED6 |
| R32_BB_CTRL_TX | 0x4000C12C | Transmit Control Register | 0x00010008 |
| R32_BB_RSSI_ST | 0x4000C130 | BB Reception Signal Quality Status | 0x00000000 |
| R32_BB_INT_EN | 0x4000C134 | BB Interrupt Enable | 0x00000003 |
| R32_BB_INT_ST | 0x4000C138 | BB Interrupt Status | 0x00000000 |

### 19.4.2 Function and description

Data whitening, transmission and reception of AA, transmission and reception of CRC initial value, and communication channels can be configured.

Configure the transmitted data:

(1) Configure the address of the transmitted access.
(2) Configure the initial value of the transmitted CRC.
(3) Configure the communication channels of the transmitted data.

Configure the received data:

(1) Configure the address of the received access.
(2) Configure the initial value of the received CRC.
(3) Configure the communication channel of the received data.

## 19.5 AES Module

### 19.5.1 Register description

Table 19-4 List of registers related to AES module

| Name | Offset address | Description | Reset value |
|---|---|---|---|
| R32_AES_CTRL_CCMMOD | 0x4000C300 | AES Mode Register | 0x00000060 |
| R32_AES_CCMINT_EN | 0x4000C304 | AES Interrupt Register | 0x00000000 |
| R32_AES_CCMVT_INIT0 | 0x4000C308 | AES CCM Mode Initial Vector Value | 0x00000000 |
| R32_AES_CCMVT_INIT1 | 0x4000C30C | AES CCM Mode Initial Vector Value | 0x00000000 |
| R32_AES_PKT_ CNT0 | 0x4000C310 | AES CCM Mode Data Packet Count Value | 0x00000000 |
| R32_AES_PKT_CNT0 | 0x4000C314 | AES CCM Mode Data Packet Count Value | 0x00000000 |
| R32_AES_DATA0 | 0x4000C318 | AES Data Register | 0x00000000 |
| R32_AES_DATA1 | 0x4000C31C | AES Data Register | 0x00000000 |
| R32_AES_DATA2 | 0x4000C320 | AES Data Register | 0x00000000 |
| R32_AES_DATA3 | 0x4000C324 | AES Data Register | 0x00000000 |
| R32_AES_KEY0 | 0x4000C328 | AES Encryption Key Register 0 | 0x00000000 |
| R32_AES_KEY1 | 0x4000C32C | AES Encryption Key Register 1 | 0x00000000 |
| R32_AES_KEY2 | 0x4000C330 | AES Encryption Key Register 2 | 0x00000000 |
| R32_AES_KEY3 | 0x4000C334 | AES Encryption Key Register 3 | 0x00000000 |
| R32_AES_KEY4 | 0x4000C338 | AES Encryption Key Register 4 | 0x00000000 |
| R32_AES_KEY5 | 0x4000C33C | AES Encryption Key Register 5 | 0x00000000 |
| R32_AES_KEY6 | 0x4000C340 | AES Encryption Key Register 6 | 0x00000000 |
| R32_AES_KEY7 | 0x4000C344 | AES Encryption Key Register 7 | 0x00000000 |
| R32_AES_RAND0 | 0x4000C348 | AES Random Value | 0x96906220 |

| | | | |
|---|---|---|---|
| R32_AES_RAND1 | 0x4000C34c | AES Random Value | 0x8A3BBF80 |
| R32_AES_RAND2 | 0x4000C350 | AES Random Value | 0xB5625304 |
| R32_AES_RAND3 | 0x4000C354 | AES Random Value | 0x2A43B7E8 |

The specific applications should be based on the BLE protocol stack library and refer to the provided BLE application example.

# Chapter 20 Parameters

## 20.1 Absolute maximum value

Critical value or exceeding the absolute maximum may cause chips to work abnormally or even be damaged.

Table 20-1 Absolute maximum parameters

| Name | Description of parameters | | Min. | Max. | Unit |
|---|---|---|---|---|---|
| TA | Ambient temperature at work | Fsys > 32MHz | 0 | 70 | °C |
| | | Fsys <= 32MHz | -40 | 85 | °C |
| TS | Ambient temperature during storage | | -40 | 105 | °C |
| VIO33 | System power supply voltage (V1033 is connected to the power, GND to ground) | | -0.4 | 3.9 | V |
| VIO | Voltage on the input or output pins | | -0.4 | VIO33+0.4 | V |
| VIO5 | Support 5V withstand voltage on the input or output pins | | -0.4 | 5.5 | V |
| VDCI | Voltage on VDCID/VDCIA pins (if external DC-DC is used) | | -0.4 | VIO33+0.4 | V |
| VXCK | PA10/PA11 voltage after X32MI/X32M0/LSE is enabled | | -0.3 | 1.4 | V |

## 20.2 Electrical parameters

Test conditions: TA = 25°C, VIO33 = 3.3V, Fsys = 16MHz.

Table 20-2 Electrical Parameters

| Name | Description of parameters | | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| VIO33 | System supply voltage | | 2.2 | 3.3 | 3.6 | V |
| $ICC_8$ | Straight-through mode static supply current | Fsys=8M | 1.3 | 1.5 | | mA |
| $ICC_{16}$ | | Fsys=16M | 1.6 | 1.9 | | mA |
| $ICC_{32}$ | | Fsys=32M | 2.3 | 2.8 | | mA |
| $IDDC_8$ | Static supply current after internal DC-DC is enabled | Fsys=8M | 0.8 | 1.1 | | mA |
| $IDDC_{16}$ | | Fsys=16M | 1.0 | 1.3 | | mA |
| $IDDC_{32}$ | | Fsys=32M | 1.5 | 1.8 | | mA |
| VIL | GPIO low level input voltage | | 0 | | 0.9 | V |
| VIH | GPIO high level input voltage | | 2.0 | | VIO33 | V |
| VIL5 | Support GPIO low level input voltage with 5V withstand voltage | | 0 | | 0.9 | V |
| VIH5 | Support GPIO high level input voltage with 5V withstand voltage | | 2.0 | | 5.0 | V |
| VOL | Low level output voltage (5mA/20mA draw current) | | 0 | 0.3 | 0.4 | V |
| VOH | High level output voltage (5mA/20mA output current) | | VIO33-0.4 | VIO33-0.3 | VIO33 | V |
| IIN | Input current of GPIO floating input end | | -3 | 0 | 3 | uA |
| IUP | Input current of the input end of the GPIO's built-in pull-up resistor | | 25 | 60 | 90 | uA |
| IDN | Input current to the input end of the GPIO's built-in pull-down resistor | | -90 | -60 | 25 | uA |
| Vref | Voltage at VINTA pins (ADC reference voltage) | | 1.03 | 1.05 | 1.07 | V |
| Vdci | Voltage at VDCID pins after DC-DC is enabled | | 1.33 | 1.37 | 1.43 | V |

## 20.3 Power consumption in the low-power mode

Test conditions: TA = 25°C, VIO33 = 3.3V, Fsys = 16MHz.

Table 20-3 Low power parameters (for reference only, temperature dependent)

| Low-power mode | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|
| Idle mode, turn on the clock combination of each module | 1.15 | 1.2 | 1.6 | mA |
| Idle mode, turn off all sleep clocks | | 1.15 | | mA |
| Pause mode, FlashROM standby | | 470 | | uA |
| Pause mode, FlashROM disabled | | 420 | | uA |
| Sleep mode, various combinations, refer to Table 5-3 | | 0.6～2.6 | | uA |
| Sleep mode, PMU + core + RAM2K, GPIO wake up, no RTC | | 0.6 | | uA |
| Power-down mode, various combinations, refer to Table 5-3 | | 0.2～1.5 | | uA |
| Power-down mode, only PMU, reset after GPIO wake-up, no RTC | | 0.2 | | uA |

Table 20-4 Current of each module (for reference only, temperature dependent)

| Name | Description of parameters | | | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|---|
| $I_{DD(RAM2K)}$ | RAM2K: 2KB SRAM | | | | 0.3～0.5 | | uA |
| $I_{DD(RAM14K)}$ | RAM14K: 14KB SRAM | | | | 0.8～1.2 | | uA |
| $I_{DD(LSI)}$ | Internal LSI oscillator | | | | 0.3 | | uA |
| $I_{DD(LSE)}$ | External LSE oscillator | | | | 0.4 | | uA |
| $I_{DD(HSI)}$ | Internal HSI oscillator | | | | 160 | | uA |
| $I_{DD(HSE)}$ | External HSE oscillator | | | 100 | 200 | 300 | uA |
| $I_{DD(BM)}$ | Battery low-voltage monitoring BM module (in pause mode) | | | | 3 | | uA |
| $I_{DD(PLL)}$ | Internal PLL oscillator | | | | 1.0 | | mA |
| $I_{DD(ADC)}$ | ADC module | | | | 0.27 | | mA |
| $I_{DD(TKEY)}$ | TouchKey module | | | | 0.08 | | mA |
| $I_{DD(TS)}$ | Temperature sensor (TS) module | | | | 0.1 | | mA |
| $I_{DD(USB)}$ | USB module | | Non-transmission status | 1.2 | 1.6 | 2.0 | mA |
| | | | Transmission status | | 3 | | mA |
| $I_{DD(ETH)}$ | ETH module | | Non-transmission status | 2.2 | 2.6 | 3.0 | mA |
| | | | Continuous transmission: rated drive | 75 | 100 | | mA |
| | | | Continuous transmission: energy-saving drive | 56 | 75 | | mA |
| $I_{DD(BLE)}$ | BLE | Reception | Straight through power supply | | 11 | | mA |
| | | | Enable DC-DC | | 6 | | mA |
| | | -20dBm | Straight through power supply | | 4 | | mA |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | transmitted power | Enable DC-DC | | 2 | | mA |
| | | 0dBm transmitted power | Straight through power supply | | 10 | | mA |
| | | | Enable DC-DC | | 5 | | mA |
| | | +3dBm transmitted power | Straight through power supply | | 16 | | mA |
| | | | Enable DC-DC | | 8 | | mA |

## 20.4 Clock source

Table 20-5 High-speed Oscillator HSI and HSE

| Name | Description of parameters | | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| $F_{HSI}$ | Internal HSI oscillator frequency | | | 32 | | MHz |
| $A_{HSI}$ | HSI oscillator accuracy | TA=-40°C~85°C | | 1.8 | 3.0 | % |
| | | TA=-5°C~70°C | | 1.1 | 1.8 | % |
| | | TA=10°C~45°C | | 0.8 | 1.3 | % |
| $T_{SUHSI}$ | Internal HSI oscillator is activated to stabilization time | | | 0.2 | 1 | uS |
| $F_{HSE}$ | External HSE oscillator frequency (20ppm when wireless communication) | | | 32 | | MHz |
| $T_{SUHSE}$ | External HSE oscillator is activated to stabilization time | | 200 | 500 | 1200 | uS |

Table 20-6 Low-Speed Oscillator LSI and LSE

| Name | Description of parameters | | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| $F_{LSIR}$ | Internal LSI oscillator frequency (before calibration) | | 20K | 32K | 48K | Hz |
| $F_{LSI}$ | Internal LSI frequency (when the application software runs after calibration) | | 32700 | 32768 | 32836 | Hz |
| $A_{LSI}$ | LSI oscillator accuracy (after software calibration) | TA=-40°C~85°C | | 0.2 | 0.9 | % |
| | | TA=0°C~60°C | | 0.1 | 0.4 | % |
| $T_{SULSI}$ | Internal LSI oscillator is activated to stabilization time | | | 40 | 100 | uS |
| $T_{SULSE}$ | External LSE oscillator is activated to stabilization time | | 150 | 450 | 1000 | mS |

Table 20-7 PLL features

| Name | Description of parameters | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|
| $F_{PLL}$ | PLL frequency multiplication output clock (CK32M*15 times) | | 480 | | MHz |
| $T_{PLLLK}$ | PLL phase lock time | | 1 | 3 | mS |

## 20.5 Time Parameters

Test conditions: TA = 25°C, VIO33 = 3.3V, Fsys = 6.4MHz.

Table 20-8 Time parameters

| Name | Description of parameters | | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| $T_{VR}$ | VIO33 voltage rising time (CH410 can be added externally if slower) | | 1 | | 50000 | uS |
| $T_{rpor}$ | Reset delay after power-on reset RPOR | | 11 | 15 | 20 | mS |
| $T_{rst}$ | RST# effective signal width | | | 100 | | nS |
| $T_{mr}$ | Reset delay after external reset MR | | 2 | 8 | 18 | uS |
| $T_{sr}$ | Reset delay after software reset SR | | 2 | 8 | 18 | uS |
| $T_{wtr}$ | Reset delay after watchdog reset WTR | | 10 | 12 | 18 | uS |
| $T_{WAK}$ | Wake-up time when exiting from low-power state | Idle mode | 0.6 | 0.8 | 1.6 | uS |
| | | Pause mode, ROM standby | 0.7 | 1.0 | 2.0 | uS |
| | | Pause mode, ROM disabled | 9 | 11 | 15 | uS |
| | | Sleep mode | 300 | 330 | 400 | uS |
| | | Power-down mode | 350 | 380 | 450 | uS |

*Note:*
*The delay parameters in the above table are based on the multiple of Tsys. Lowering the main frequency will increase the delay.*
*The delay parameters in the above table are based on the use of the internal HSI clock sources. If the external HSE clock sources are used during sleep, the delay parameter TWAK of the pause mode/sleep mode/power-down mode in the table will be increased by about 1.5mS (activated to stabilization).*

## 20.6 Other parameters

Test conditions: TA = 25°C, VIO33= 3.3V, Fsys = 16MHz.

Table 20-9 Other parameters

| Name | Description of parameters | | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| $R_{TS}$ | Measuring range of temperature sensor | | -40 | | 90 | ℃ |
| $A_{TSC}$ | Measurement error of temperature sensor after software calibration | | | ±6 | | ℃ |
| $T_{FRER}$ | Operation time of single sector erase of Flash-ROM | | 1.1 | 1.4 | 2.4 | mS |
| $T_{FRPG}$ | Operation time of single word programming of Flash-ROM | | 22 | 27 | 36 | uS |
| $N_{EPCE}$ | Number of Flash-ROM erases, Erase/ program cycle endurance | 5~45°C | 100K | 800K (non-guaranteed) | | times |
| | | -40~85°C | 30K | 100K (non-guaranteed) | | |
| $T_{DR}$ | Data retention capacity of Flash-ROM | | 10 | | | years |
| $V_{ESD}$ | ESD withstand voltage on I/O input or output pins | Antenna (ANT) | 2K | 4K (non-guaranteed) | | V |
| | | I/O pin: PA and PB | 4K | 6K (non-guaranteed) | | V |

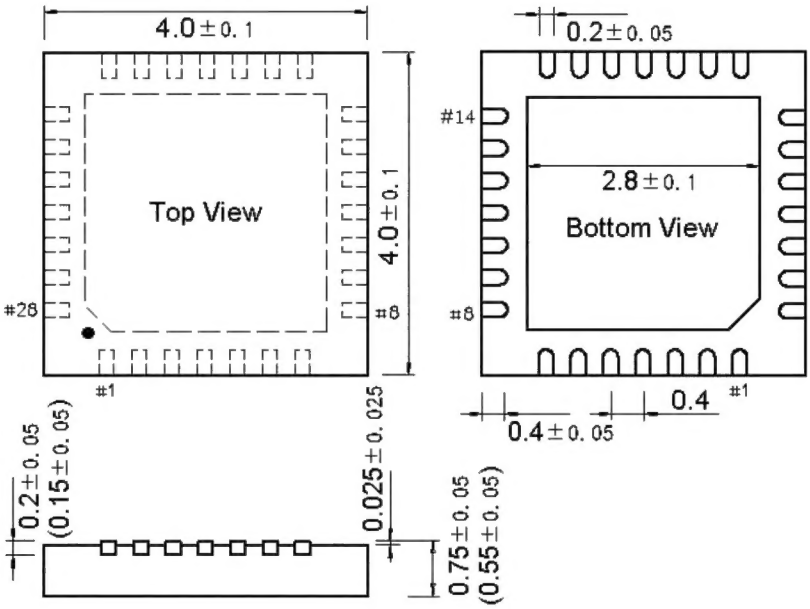
# Chapter 21 Package

## Chip package

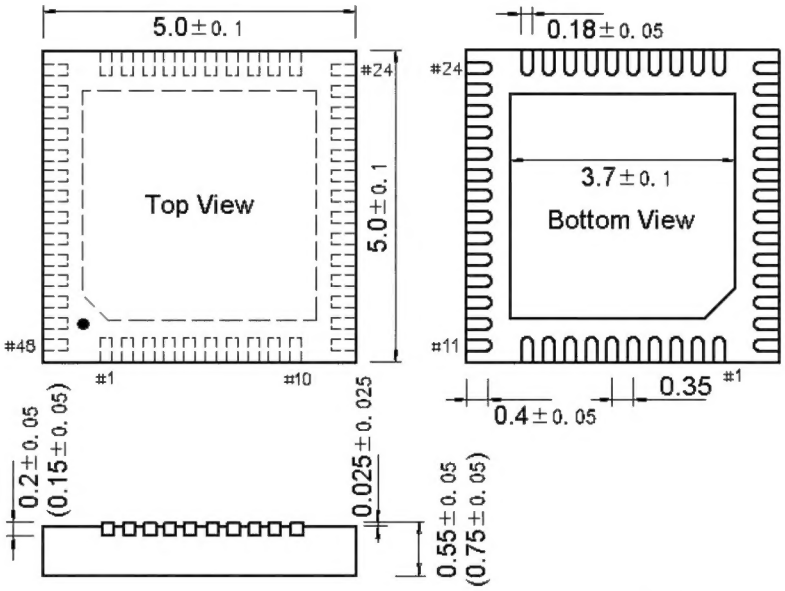
| Package | Width of plastic | Pitch of pin | | Package description | Order model |
|---|---|---|---|---|---|
| QFN28 | 4*4mm | 0.4m | 15.7mil | Square leadless 28-pin | CH579F |
| QFN48 | 5*5mm | 0.35m | 13.8mil | Square leadless 48-pin | CH579M |

*Note: The unit of dimensions indicated is mm. The distance between centers of the pins is equal to the nominal value. The dimensional error other than this is not more than ±0.2mm.*

QFN28




QFN48




Default is 0.55

# Chapter 22 Revision Records

| Version | Date | Description |
|---|---|---|
| V0.9 | May 09, 2018 | Release the original version |
| V1.0 | September 28, 2018 | Release the official version |
| V1.1 | October 23, 2018 | Add RB_PWR_MUST_0010 to Section 5.2; change (1) in Section 6.4.4; change R8_ADC_CFG in Section 15.2; change $N_{EPCE}$ in Section 20.6; change the typo of R16_OSC_CAL_CNT in Section 6.3; change the typo in Table 17-3 |
| V1.2 | March 04, 2019 | Change VSW to inductance in Section 1.2; change R16_POWER_PLAN in Section 5.2; delete "low voltage detection in sleep mode" in Section 5.3; change R16_CLK_SYS_CFG in Section 6.3; delete RB_CHG_CNT in Section 16.2 and Section 16.3 |
| V1.3 | April 13, 2019 | Add WFE to Table 5-2; adjust the parameters in Table 20-9 |
| V1.4 | July 26, 2019 | Part of control bits in R16_PIN_ANALOG_IE in Section 7.4; R8_SLV_RD_DATA address in Section 14.1 |
| V1.5 | April 15, 2020 | For RF, only BLE is reserved. Steps are modified in Section 6.4.4. Modify the typo of low speed in Section 17.2.2. |
| V1.6 | December 02, 2020 | Modify the typo, modify R16_CLK_SYS_CFG in Section 6.3, add description of the count mode for timers. Modify the default value in Table 2-3. Add $T_{vr}$ in Table 20-8. Add Section 10.5 about SPI timing. |
| V1.7 | August 11, 2021 | Note that thickness of package is 0.75mm by default. |
| V1.8 | September 24, 2021 | Update HIS oscillator accuracy $A_{HSI}$. Update packages, with the size added. |
| V1.9 | September 30, 2022 | R16_AUX_POWER_ADJ If the write operation requires 16 bits, delete the filter information |
| V2.0 | October 19, 2022 | Update document format, package thickness default 0.55mm, update sleep current reference value range |
| V2.1 | November 30, 2022 | Update reference voltage Vref, update Fsys > 32MHz operating temperature range |
| | | |